A Connectionist Model of Sentence Comprehension and Production

Douglas L. T. Rohde

CMU-CS-02-105
March 5, 2002

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy

Thesis Committee
Dr. David C. Plaut, Chair
Dr. James L. McClelland
Dr. David S. Touretzky
Dr. Maryellen C. MacDonald

This research was supported by an NSF Graduate Fellowship,
NIMH NRSA Grant 5-T32-MH19983 for the study of Computational and Behavioral Approaches to Cognition,
and NIH Program Project Grant MH47566 (J. McClelland, PI)

Keywords: language, connectionist, neural network, learning, sentence, comprehension, production

Abstract

The most predominant language processing theories have, for some time, been based largely on structured knowledge and relatively simple rules. These symbolic models intentionally segregate syntactic information processing from statistical information as well as semantic, pragmatic, and discourse influences, thereby minimizing the importance of these potential constraints in learning and processing language. While such models have the advantage of being relatively simple and explicit, they are inadequate to account for learning and validated ambiguity resolution phenomena. In recent years, interactive constraint-based theories of sentence processing have gained increasing support, as a growing body of empirical evidence demonstrates early influences of various factors on comprehension performance. Connectionist networks are one form of model that naturally reflect many properties of constraint-based theories, and thus provide a form in which those theories may be instantiated.

Unfortunately, most of the connectionist language models implemented until now have involved severe limitations, restricting the phenomena they could address. Comprehension and production models have, by and large, been limited to simple sentences with small vocabularies (cf. St. John & McClelland, 1990). Most models that have addressed the problem of complex, multi-clausal sentence processing have been prediction networks (cf. Elman, 1991; Christiansen & Chater, 1999a). Although a useful component of a language processing system, prediction does not get at the heart of language: the interface between syntax and semantics.

The current thesis focuses on the design and testing of the Connectionist Sentence Comprehension and Production (CSCP) model, a recurrent neural network that has been trained to both comprehend and produce a relatively complex subset of English. This language includes such features as tense and number, adjectives and adverbs, prepositional phrases, relative clauses, subordinate clauses, and sentential complements, with a vocabulary of about 300 total words. It is broad enough that it permits the model to address a wide range of sentence processing phenomena. The experiments reported here involve such issues as the relative comprehensibility of various sentence types, the resolution of lexical ambiguities, generalization to novel sentences, the comprehension of main verb/reduced relative, sentential complement, subordinate clause, and prepositional phrase attachment ambiguities, agreement attraction and other production errors, and structural priming.

The model is able to replicate many key aspects of human sentence processing across these domains, including sensitivity to lexical and structural frequencies, semantic plausibility, inflectional morphology, and locality effects. A critical feature of the model is its suggestion of a tight coupling between comprehension and production and the idea that language production is primarily learned through the formulation and testing of covert predictions during comprehension. I believe this work represents a major advance in the attested ability of connectionist networks to process natural language and a significant step towards a more complete understanding of the human language faculty.

For Christine, my future wife and my inspiration, whose love is my abiding companion, and whose superior aptitude for language will forever be a thorn in my side.

Acknowledgments

Thanks are due first to my committee, Dave Plaut, Jay McClelland, Dave Touretzky, and Maryellen MacDonald, for their help in carrying out this project, and the significant insights I lifted from their own work.

Dave Touretzky has been my Black Friday defender, whose research for me shows the importance of seeking the link between behavior and the neural mechanisms of the brain. Maryellen was my frequent consultant on all things psycholinguistic, and reference to her ideas on the importance of probabilistic constraints in language processing will be an annoyingly common theme in this thesis. Jay has been a major influence on me throughout my graduate years and, having been a key pioneer of the field of connectionism, his work is the main foundation on which this project was built. Jay has the remarkable ability to take some of my best new ideas and slip them into stuff he published twenty years ago.

And, of course, Dave Plaut has had a tremendous influence not just on the design of the current model, but on my overall outlook on the study of language, computation, and the mind. I have learned from him a healthy skepticism towards bad science and an optimism towards good science. I can only hope that I've also adopted some of his ability to tell the difference.

Ted Gibson, apparently out of a sense of sheer pity, was kind enough to give me an office at MIT, where I have squatted for the past year and a half. In exchange, I have made him the most-cited author in this thesis. His work isn't bad either. I am looking forward to continuing to collaborate with him and use his copier code in the months and years to come.

Thanks are also due to the other members of TedLab—Dan Grodner, Tessa Warren, Duane Watson, Florian Wolf, and fellow squatters Timothy Desmet and John Hale. They continue to be good friends and colleagues. Dan, in particular, has helped me shape the arguments expressed here by his willingness to engage in frequent and interminable debates. He has yet to be correct, except of course when agreeing with me, but I admire the tenacity.

"The Boys" from CMU—Doug Beeferman, Adam Kalai, Neil Heffernan, Santosh Vempala, Philip Wickline, and Carl Burch—were my constant pals in Pittsburgh, and even accompanied me to Boston to continue distracting me from my work, a skill at which they excel and for which they are much appreciated. Stan Chen got stranded half way, but he'll make it eventually.

Lastly, I would like to thank my parents, Mary and Lee, and my brothers Lars and Gilbert, who have been a steady source of support and encouragement, but who probably think finishing this means I'll finally get a job. Inexpressible gratitude should also be extended to my mother for copy editing this thesis, an unenviable task that may have made her the only person other than myself to ever read the entire thing. She wishes to disavow any responsibility for these acknowledgments.

Contents

Chapter 1

Introduction

Our unique faculty for language is one of the most fascinating and perplexing of human abilities. As researchers who seek to understand it, we are enticed by the apparent ease with which nearly all people learn and use everyday language. But when we begin to delve under the surface of the human language system, its true complexity seems overwhelming. Decades of research in processing language with computers—from attempts to comprehend sentences, translate between languages, or even just to generate and transcribe speech—have all met with limited success. Our most powerful software and machines are, by and large, no match for the linguistic abilities of a typical five-year-old.

A complete theory of human language must involve an understanding of the neural mechanisms by which our minds process language, including the nature of the representations used by the system, how it learns, and what innate knowledge or structure is required for it to do so. Such knowledge will be critical for advancing the development of computational linguistics, in helping those with developmental or acquired language deficits, and in simply fulfilling our scientific and intellectual curiosity.

The study of other cognitive phenomena, including vision and memory, have benefitted greatly from the existence of other species that share human abilities in these domains. The use of invasive techniques in animal models, including extensive neural recording, in vitro study of living tissue, radioactive labeling, and induced lesions, have been a primary means by which our understanding of these domains has grown. However, because language is a uniquely human ability, we are greatly restricted in our options for directly investigating the neural mechanisms of language. Although some information can be gained through the study of naturally occurring lesions and via functional imaging, the primary empirical method we have for investigating language is the controlled elicitation, measurement, and analysis of human behavior.

But empirical data is of little use unless it can be fit into a broader framework or theory. If they are to tell us anything about language mechanisms, empirical observations must be tied to specific functions or processes. This is the role of theories and conceptual models. In addition to explaining the available empirical data, theories, if they are good ones, should enable us to make predictions about the outcomes of future experiments, provide insights that could lead to more effective computer language processing, and aid in the development of treatments for language impairments. Although the importance of articulating theories of language, or other mental processes, is well recognized in the scientific community, many researchers have avoided actually implementing their theories as explicit, testable models (Lewis, 2000b).

1.1 Why implement models?

The strongest excuses for not modeling a theory are that it is time consuming and requires a degree of technical sophistication. While, depending on the type of model, these may be valid complaints, in most cases they are far outweighed by the many benefits of implementing explicit models of cognitive phenomena.

- The constraints of implementing a model force theories to be complete, clear and detailed. Researchers may think it unnecessary to implement their theory because, to them, the theory seems perfectly clear and its predictions obvious. But English is not a precise language for describing complex or abstract processes. If a theory is only described in words, other researchers may interpret the description very differently from the author's intention, or may get the correct general idea but misinterpret the details. Unless a theory is explicit, other researchers will be unable to properly evaluate it or predict its behavior under novel conditions. Unimplemented theories may also rest on hidden assumptions that are not fully thought out or understood by the authors themselves.

- The demands of implementing a model quickly reveal flaws in the theory. Although a theory may seem well articulated, casting it in algorithmic and numeric terms can reveal hidden inconsistencies and vagueness. Thus, before a model is even evaluated, that it can be constructed is a valuable test of a theory, and is likely to inspire needed extensions or changes. Furthermore, theories of a particular aspect of language processing may lack connection to theories specific to other aspects of language processing or to the domain as a whole. But when we implement a model, we are forced to define its bounds and the representations used at its interfaces. This leads the designer to think critically about how the model fits in with other theories and with language processes at a broader level.
- Some theories must be implemented if they are to be fully understood. The behavior of complex systems cannot easily be predicted, and all but the most trivial of cognitive models will be too complex for us to simulate in our minds. If a theory involves two factors that, for a particular input, pull in opposite directions, the net result, and hence the theory's prediction, may depend on details that are only made explicit, and can only be understood, when a model is constructed. Some theories, such as the one on which the current work is based, are heavily dependent on learning. The outcome of the model depends on the environment to which it is exposed. Unless the learning process and the environment are very simple, the model must be trained and tested before its properties will be known.
- Evaluating a model against empirical data can reveal flaws in the model and, by extension, in the theory. This is, of course, the main benefit that usually comes to mind when we think of modeling. Models can be used to simulate empirical experiments. When the results match the empirical findings, the theory is strengthened. When the results do not match, it tells us that either the model is not representative of the theory or that there is a flaw in the theory. This is the potential falsifiability that some see as a requirement of any valid scientific theory.
- Evaluating a model against empirical data can reveal flaws in the data, leading to improved experimental design. All experiments are flawed. As scientists, when we design an experiment to test the effects of a particular factor, we will attempt to control for or rule out all confounding factors. But identifying and controlling for all potential confounds is simply not possible. Biases can creep in at many levels: in our choice of subjects, task design, item selection, use of filler items, testing methods, data analysis techniques, and so forth. In general, we do our best to eliminate any obvious confounds and then assume the results are reliable. But when an experimental study becomes the critical test for our model and the two happen to disagree, there is a tendency to look very closely at the details of the empirical work. Often, the problem is not the model but the data. Experimental design flaws, made evident by critical comparison with a model, can be reduced in better controlled experiments. Even if the model is ultimately proved wrong, there is still a net benefit to science.
- Implementation aids theory formation by providing further insight into the workings of the model and the importance of its various aspects. As a designer of models, I find that my models often surprise me. They don't always work the way I had intended or, if they do, it is not always for the reason I had expected. This is bound to happen with any non-trivial model, but is especially likely with models having a substantial learning component. Where the model's actual behavior is undesirable, it may mean that further constraints on the model are needed. But when the model's actual behavior is correct, it can lead to further understanding and elaboration of possible functional mechanisms behind the observed phenomena.
- Finally, modeling inspires new theoretical questions and new empirical experiments. The process of implementing and studying a model may raise questions that would not occur to us otherwise. If we see that the model is sensitive to a particular factor, we may create a new experimental design to test the sensitivity of humans to that factor. Sometimes, a single mechanism can account for a range of phenomena that, on the surface, do not appear to be related, resulting in unification and simplification of theories.

Thus, modeling is not just important for evaluating theories of complex phenomena. It plays a critical role in rendering theories more complete and explicit, improving our understanding of them, and inspiring new theoretical developments and new empirical investigations.

The primary goal of this thesis is to gain a better understanding of a particular pair of domains within language processing—sentence comprehension and production—through the development and testing of a broad-scale model of human performance.

1.2 Properties of human language processing

Before discussing the model, let us begin by considering some general characteristics of the human language processing system. The following are what I consider to be important properties of language and the way in which we use it. Some of these points are obvious, some are controversial, and the aim here is not to defend them with specific empirical findings. That may come later. The point is simply to lay out some observations on what I see as critical characteristics of human language. These observations, or positions, serve to guide my thinking about language and motivate the approach to modeling sentence processing that is adopted in this thesis.

- To begin with, language is structured. It is undeniable that all human languages have clear regularities to their structure. A language is not simply a collection of idiomatic expressions that convey unique messages, although those are an important part of many, if not all, languages. Most sentences are composed of elements, such as noun and verb phrases, that are arranged in regular and predictable patterns of relationships with one another. These patterns, if codifiable, form the syntax or grammar of the language.

  The field of modern linguistics, taking its inspiration largely from the work of Chomsky, has focused primarily on issues regarding the syntactic regularity of language. In doing so, it revealed many of the important properties of syntax, within and across languages. But, for a long time, this came at the cost of virtually ignoring many other aspects of language, including meaning and how it is mapped to structure and surface form, the statistical properties of language, and our actual, workaday comprehension and production performance.
- Language is productive. Familiar words can be combined according to the rules of grammar to produce a vast array of novel sentences that can be comprehended by other speakers of the same language, to whom the sentences are equally novel. It is this undeniable fact that was a principal motivation behind Chomsky's well-known critique of Skinner's theory of verbal behavior (Chomsky, 1959).
- Semantic and lexical constraints place tight bounds on the productivity of language. Although many of the sentences we encounter on a daily basis are novel, there are rather tight semantic and pragmatic bounds on the way words can be combined within a syntactic framework to produce reasonable sentences. Example (1a) is a perfectly valid English sentence, but when we rearrange its syntactic elements—its nouns, verbs, determiners, and prepositional phrases in this case—the resulting (1b) is grammatical but nonsense that the man on the street would probably argue is unacceptable as English.[1]

  (1a) The student believes in evolution, but he knows of another theory.
  (1b) Another theory believes of the student, but he knows in evolution.

  To give another example, one can be a *student of religion* or *reduce the deficit 20%*, but cannot be a *student of 20%* or *reduce the deficit religion*. You might argue that these generalizations are invalid not just on semantic grounds but on syntactic grounds according to a theory in which quantifiers, like *20%*, are syntactically distinct from other noun phrases. But this is largely my point—that there are many limitations to the productivity of natural language. One who constructs a generative grammar of a language and then actually tries to use it to generate sentences will probably find that it produces mainly strange and unacceptable utterances. A real generative theory must take much more into account than basic syntactic classes and how they can be arranged.
- Language is *pseudo-context-free* or *pseudo-context-sensitive*, but infinite recursion is an idealization ungrounded in any observable reality. Some like to say that there are an infinite number of possible sentences in a language, meaning that there are an infinite number of utterances that conform to the rules of the grammars that linguists design. But if we limit the definition of a sentence, as I choose to do, to those utterances that could possibly convey a useful and consistent meaning to at least a few fellow speakers of a language, even under optimal conditions, the claim of an infinite variety of sentences is simply not true. There is a finite bound to the length and complexity of

[1] One might argue that "believe of" is never grammatical in English. But, it does have valid uses, such as this one in a recently published editorial: "For we know that no matter what happens, it is the fault of our enemies, for they dare to believe of themselves what we believe of ourselves."

sentences that humans can comprehend, and thus a finite bound to the possible sentences we can create, short of inventing new words. If I were to link all of the sentences in this thesis together with *and* or some other suitable conjunction, the result would not be another sentence. It would be an abomination. I'm not arguing that there aren't a vast number of possible sentences in English,[2] just that there aren't an infinite number.

Many researchers have been attracted to the idea of language as context-free because of the relative ease with which the basic grammars of most human languages can be described by systems of context-free rewriting rules. The existence of repeatable embeddings can be easily described in such systems using a symbol whose possible productions include itself. It is tempting, though not necessarily justified, to make the assumption that it is valid to do this ad infinitum, and that any sentences produced by an arbitrary number of self embeddings, or any other productions of the grammar, are also part of the natural language being characterized. Such sentences may be part of the theoretical language defined by the grammar, but they are not necessarily part of the natural language in which we are interested. Much of the debate in computational linguistics has been about whether human languages are really context-free or, perhaps, belong to the superset of context-sensitive languages.

Traditional finite-state grammars, by contrast, have not been thought to be sufficient for characterizing natural language. This is because it is usual to conceive of and describe such grammars using state transition diagrams. Center embeddings, beyond the first level, cannot be described conveniently using state transition diagrams. To do so would require duplicating extensive amounts of structure for each level allowed. This seems so unwieldy that most researchers have jumped straight to context-freedom and have thus embraced infinite recursion as a property of natural language. However, there is an alternative, which is to seek out a more powerful means for describing finite-state grammars than the state transition diagram.

Although all regular languages can be described by a state transition diagram, it is not necessarily the most convenient way to do so. One alternative is to use something exactly like a context-free rewriting system, but with a bound placed on the depth of the production tree or on the number of self-embeddings allowed. The set of languages producible by a context-free grammar with a finite limit on the depth of self-recursion is equivalent to the regular languages, describable by finite-state transition diagrams.[3] But because this method uses symbol rewriting rules, it can quite easily describe most basic natural language grammars, but it would not be convenient for describing the limited context-sensitivity that shows up in, for example, Dutch *scrambling*. But there is no reason one can't invent a new formalism for describing finite-state grammars that permits easy notation of both self-embedding and scrambling. In this way, we can reap the benefits of more powerful notation, without committing ourselves to the absurd stance that certain 1-million-word sentences are grammatical, whereas others are not. For this reason, I think of natural language as pseudo-context-free or pseudo-context-sensitive. It contains some of the hallmarks of context-free or context-sensitive grammars, but does not permit arbitrary complexity. My perspective here, as elsewhere, is by no means unique, and arguments for the finiteness of natural language have been around for a long time (Reich, 1969).

- Syntax is interesting and important; but the goal of language is to convey meanings, and semantics is a domain of subtle distinctions and gray areas. Although it is possible to develop a reasonably comprehensive theory of syntax based on discrete categories and rules, semantics is, by nature, much fuzzier. There is an infinite range of variation to the types of objects, entities, actions, and ideas in the world. When we assign words to these concepts, we are forced to create categorizations and establish sometimes arbitrary distinctions. But there are often profound similarities between concepts we refer to by different names, and profound differences between concepts we refer to by the same name. Semantic knowledge is not easily described or codified. Consider all the wines produced in the world today. They can be roughly divided into classes on the basis of their ingredients and how they are made. But each one is different. Each has its own characteristic flavor, texture, aroma, and color. These together, or our sense of them, make up the concept of that individual wine. Even experts struggle to classify and describe them, resorting to combinations of metaphorical adjectives like *austere*, *flinty*, and *supple*.

  Effective communication often requires not just that words or categories be conveyed, but that the underlying concepts, or an approximation to them, be communicated. After reading (2a), you may have the image of a dog that is big and lovable, but somewhat disgusting. But (2b) was intended to convey the image of a little, obnoxious, hyperactive dog. Although words themselves are vague and often polysemous, skilled comprehension is about

[2] This thesis contains approximately 10,000 sentences, and I like to think that at least most of them are novel. I'm sure the previous one was, but I'm not certain about this one.

[3] I have found a truly marvelous proof of this theorem which this footnote is too small to contain.

deriving precise meanings from the way in which they are used together in a sentence or discourse, and skilled production is about conveying precise meanings by selecting and arranging words in just the right way.

(2a) The dog slobbered all over its favorite stuffed bear.
(2b) The dog yipped at my feet but I resisted the temptation to give it a kick.

- Lexical ambiguities abound in natural language, but are rarely even noticeable. Aside from subtle distinctions among similar word senses, many words in English have multiple, unrelated or distantly related senses and can also play more than one syntactic role. A simple word like *fast*, for example, can serve as a noun, verb, or adjective, as illustrated in (3). Of the 1,000 most common verb forms in the Wall Street Journal and Brown corpora of the Penn Treebank (Marcus, Santorini, & Marcinkiewicz, 1993), 22% can also serve as nouns. And of the top 1,000 noun forms, 29% can serve as verbs. In addition to this, the majority of nouns and verbs have more than one distinct sense. The verb *to crash* might refer to a physical collision, a plummeting of the stock market, junkies coming off a high, the emission of thunder, or arriving at a party without an invitation. For the most part, we are able to use context to resolve sense ambiguities and we may not even be aware that a single word plays a variety of unrelated or remotely related roles. Consider, for example, (4), borrowed from Johnson-Laird (1983), which is not too difficult to understand even though every content word is ambiguous. One can even construct sentences, such as (5), that are globally ambiguous, although it is not obvious to us on first reading. The usual first interpretation for this sentence is that an old man is boating, but it could also mean that old people are typically found manning boats.

(3) The activists vowed to hold fast to their plan to fast for 90 days, but the allure of the fast food drive-through was overwhelming so their fast was soon at an end.
(4) The plane banked just before landing, but then the pilot lost control. The strip on the field runs for only the barest of yards and the plane just twisted out of the turn before shooting into the ground.
(5) The old man boats.

- Syntactic ambiguities also abound in natural language, but most go completely unnoticed as well. Temporary syntactic ambiguities are quite common in everyday speech, but we rarely detect their presence except in the few cases in which we are *garden-pathed* by an unusually deceptive sentence. Sentences (6a)–(6b), for example, are all perfectly comprehensible. And if we were to hear just one, we would probably not realize that "*If I find you*" is temporarily ambiguous and *you* could serve as the subject of a sentential complement, the indirect object in a ditransitive, or as the direct object in a simple transitive.

(6a) If I find you cheated me, you'll be sorry.
(6b) If I find you the answer, will you stop asking for help?
(6c) If I find you, then you're it.

Our ability to resolve most short-term ambiguities of this sort suggests that we are capable of temporarily maintaining several possible interpretations of a sentence (Pearlmutter & Mendelsohn, 1999; Gibson & Pearlmutter, 2000). The fact that we are sometimes garden-pathed, especially when one reading seems particularly likely, suggests that we are capable of only limited parallelism of this sort. Because the strength of an interpretation often seems to depend on its likelihood given the constraints provided by frequency and other factors, which will be discussed shortly, I think of our ability to represent multiple parses as *graded parallelism*. Only a limited number of interpretations are maintained and each has a strength associated with it. Very unlikely interpretations may be discarded and, if a surviving interpretation turns out to be correct, the ease of committing to that interpretation will depend on its strength relative to those of the other interpretations.

- The meaning of a sentence is not just the sum of its parts. It is often pointed out that one cannot simply add up the meanings of the words in a sentence to form the meaning of the whole. But neither is the meaning of a sentence always the straightforward composition of its parts. The meanings of words and phrases are frequently determined by their context, both intra- and extra-sentential. An English dictionary cannot be combined with a textbook of English grammar to produce a reasonable comprehension system. A system built in this way would probably fail to appreciate that, in (7a), the baby is settled down for an all-too-brief nap, while in (7b), canis mortuus est.

(7a) We finally put the baby to sleep.
(7b) We finally put the dog to sleep.

- Comprehension is sensitive to many factors, including lexical and structural frequency biases, semantic and pragmatic influences, and context (MacDonald, Pearlmutter, & Seidenberg, 1994a). As we'll see plenty of evidence for in Chapter 3, a variety of factors can have an early and important effect on the sentence comprehension mechanism. These effects can impact the speed or ease of processing, as well as the ultimate interpretation. It is a combination of these factors which, under most circumstances, make lexical and syntactic ambiguities relatively easy to resolve.

- Syntactic interpretation is often dependent on semantics. Proper parsing of a sentence may be dependent on and guided by its semantic content. In (8a) and (8b), taken from Taraban and McClelland (1988), the final prepositional phrase clearly serves as an instrument of the action in the one case, but the possession of the *cop* in the other. Similarly, most readers will conclude that the dog found the slippers in (9a), but the slippers were just next to the dog in (9b). In this case, the distinction, which would probably be viewed as syntactic under most theories, is not due to a particular noun/verb pairing, as before, but to the differing properties of the specific *dog* instances.

  (8a) The spy saw the cop with binoculars.
  (8b) The spy saw the cop with a revolver.
  (9a) The slippers were found by the nosy dog.
  (9b) The slippers were found by the sleeping dog.

- The meaning of a word, phrase, or sentence can depend on the context in which it is placed. A simple example of this, playing off two of the senses of the adjective *nice*, is given in (10a) and (10b). In the first case, *nice* means kind or well-tempered, while in the second it means high-quality or well-performing.

  (10a) My horse likes to buck, but you've got a nice horse.
  (10b) My horse is old and slow, but you've got a nice horse.

- Meaning can be carried by prosody, or the tone of voice. In everyday conversation, a substantial amount of information is conveyed not just by the words that are uttered, but by how they are uttered. Depending on how I delivered it to you, sentence (11) could take on a variety of meanings. If I placed stress on *how*, this is not really a question, but an indication of the fact that I cannot believe the thesis was as long, or short, as you just said it was. If I stress *long*, it is a question that seeks to elicit the length of the thesis, as opposed to some other information about it. If I stress *was*, it suggests that the length of the thesis has changed and I want to know its original length. If I stress *thesis*, it suggests you have told me the length of something else, but I prefer to know the length of the thesis. This is an example of contrastive stress, but other prosodic markers can also influence sentence meaning. If (12) is spoken with no pause, it seems to clearly mean that I saw you while you were walking in the park. But if a pause is placed after *you*, it can be taken to mean that I saw you while I was walking in the park.

  (11) How long was the thesis?
  (12) I saw you walking in the park.

- We are tolerant of syntactic or semantic anomalies. Our sentence comprehension mechanism does not break down when it encounters a syntactic violation or other mistake. We are generally able to get past these problems and still divine the speaker's true intention. This is illustrated to some extent by one of my favorite G. W. Bushisms, (13). Although this sentence contains an agreement error and some horribly mixed metaphors, it still seems possible to divine the President's intended message.

  (13) Families is where our nation finds hope, where wings take dream.

- Over time, languages tend to become regularized. When languages collide, inconsistent pidgins, over a generation, become consistent creoles (Bickerton, 1984). At a slower pace, regularizations tend to occur in the inflections and spellings of words within a language. This is especially true of low-frequency words. Only common irregularities tend to survive (Hare & Elman, 1995). Does this mean that we process language according to rules? Not necessarily. Only that we have an easier time dealing with regularities than we do with exceptions.

- We are not taught to parse. Parsing sentences into syntactic structures is not something most language users are shown how to do. In fact, it isn't even clear that we do parse, in the sense of constructing an explicit and complete representation of the structure of a sentence in the course of comprehending it. The only language tasks that we really know humans are able to perform are those we can directly observe, including comprehension, production, repetition, and so forth. That mapping from an acoustic signal to a message occurs via deep structure, or some other hierarchical syntactic representation, is merely a convenient assumption—a pervasive and often useful one, but an assumption none the less. In advancing our understanding of the human language processing system, we may do well to question this assumption.

- We are great users of language, but we are terrible linguists. Nearly all of us reach fluency in at least one language with very little direct instruction. And yet, abstract linguistic constructs and the principles that govern them do not come naturally to us. Most language users who have yet to take a course in grammar probably realize that there are such things as words and that most of them fall into the distinct classes of nouns, verbs and adjectives. But they probably couldn't articulate the fact that there are elements of their language such as those we refer to as prepositional phrases, relative clauses, mass nouns, modal verbs, reflexive verbs, gerunds, unaccusatives, or the subjunctive, unless you were to ask very leading questions. Until it occurred to me quite recently, I could not have told you that there are a small number of count nouns, like church, school, prison, and, in England, hospital, that can be used in the singular without a determiner. How is it that we are able to use these things on a daily basis, yet most of us don't realize they exist and even professional linguists struggle for years to discover the principles underlying their use.

  If we really do possess "knowledge" of these principles, it is certainly hidden deep inside of us. Is it reasonable to think that such knowledge could be encoded in our minds in the form of explicit rules and yet be totally inaccessible to our consciousness? Or, perhaps, do those principles not exist in the form of rules, but merely in the form of procedures and processes, like the neural pathways that have made me a better skier through practice, without blessing me with the ability to accurately describe those skills or pass them on to others. It is true that my skiing faculty exhibits consistent, rule-like behavior. When one ski slips, weight is transferred to the other leg. When I want to initiate a turn, my body does a consistent sequence of operations, like tilting my feet, leaning and twisting in a certain way. But the rules that govern this are *implicit* in the operations of my reflexes and higher-level motor control pathways. To characterize such a system as one based upon the knowledge of rules seems absurd.

- A substantial portion of the human brain is involved in language processing. Few would question that using language is one of the most complex and important things that we do. If 5% of the cortex were involved in sentence processing, that would mean there are roughly one billion neurons working on the problem.

- We can easily learn hundreds of thousands of words and phrases. It has been estimated that the average college graduate knows around 60,000 words (Pinker, 1994; Crystal, 1997). But if one were to include all of the senses we attribute to these words, as well as idiomatic expressions with their own unique meanings, and the proper names with which we are familiar, the number of mappings we know and use between linguistic signs and their meanings would certainly be in the hundreds of thousands. The brain is obviously quite well designed for learning and accessing such mappings.

- In contrast to the apparent ease with which we acquire vast lexica, we are embarrassingly bad when it comes to syntax. Sure we can comprehend most of the sentences encountered on a daily basis with little difficulty (that is apparent to us, anyway). But this is by design. By and large, sentences are not produced by a reasonable speaker or writer if she can't understand them herself or doesn't expect others to understand them. But as soon as we go somewhat beyond the limit of the syntactic complexity that we normally encounter, our comprehension abilities rapidly break down. To those who don't study such sentences for a living, example (14), which contains two center-embedded object-relative clauses, stops making sense around the second verb. Sentences of this type are easier to understand when strong semantic constraints are placed on the possible roles that each constituent can play (Stolz, 1967). But when we are forced to let syntax be our guide, multiply embedded structures are very confusing.

  (14) The dog the cat the mouse bit chased ran away.

  Various researchers have articulated slightly different rules to predict the limit of our ability to handle difficult syntax. Several of these equate complexity with the maximum number of incomplete syntactic dependencies at any point in the sentence (Yngve, 1960; Miller & Chomsky, 1963; Bever, 1970; Kimball, 1973; Gibson, 1991). Based on empirical observations, our limit seems to be about three. Lewis (1993, 1996) suggests a variant of this, which is that complexity is proportional to the maximal number of incomplete dependencies of the same kind. In this case, the limit is closer to two.

So here, in a nutshell, is one of the most perplexing things about human language processing. We have a huge number of neurons involved in language, which enable us to learn hundreds of thousands of mappings between linguistic surface forms and their meanings. And yet, we cannot handle more than two or three center-embeddings. This tells me that the architecture of our language processing system must be one that is specialized for acquiring associative mappings, but one with only a modest capability for dealing with complex, combinatorial structure.

1.3 Properties of symbolic models

The most predominant and the most explicit models of sentence processing have, until recently, been based on symbolic architectures. Symbolic systems are marked by the use of explicit rules that operate on abstract variables or symbols.[4] The symbol rewriting grammars in which context-free languages are specified are inherently symbolic, as are most computer programming methods. The principal attraction of symbolic notation for language processing is the ease with which it can be used to represent abstract syntactic constructs and their relationships. For example, one can simply specify that noun phrases are represented by the symbol NP and verb phrases by the symbol VP and a sentence, S, can be composed of an NP followed by a VP. Knowledge of such constructs, and the rules for operating on them, can be built directly into a symbolic model.

One of the main reasons for favoring symbolic language processing models is that syntax can be easily represented and manipulated in such a system. But they have other important advantages. In particular, symbolic models are frequently quite simple and explicit. They are relatively transparent in their operation and tend to provide a high degree of explanatory clarity. We can easily examine not just the overt behavior of such models, but their internal states and representations as well. This enables us to design them with the goal of achieving a certain behavior on a known task, or predict their behavior on a novel task. Because of their reliance on abstract symbols or variables, generalization can be quite robust in symbolic models. If a rule or property applies to one noun phrase, it could just as easily apply to all of them. This is important in accounting for the productivity of language.

Because they are so well suited to manipulating syntactic structure, symbolic models of sentence processing have mainly been applied to the task of parsing, which only by supposition is necessary for natural language processing. To address comprehension and production, on the other hand, we must have a way to describe sentence representations at the semantic level as well. Accounting for semantics is where symbolic models begin to run into trouble. I have asserted that semantics is a domain of subtle distinctions and gray areas. Semantic concepts are not so easily classified and cannot easily be represented or manipulated using abstract variables. Attempts have been made to approximate semantic spaces using discrete classifications, but close approximations require a great deal of complexity, which threatens to reduce clarity, predictability, and the possibility of generalizing.

Accounting for the empirically attested graded effects of multiple constraints in online sentence processing is also difficult in a rule-based model. Rules are generally phrased in terms of discrete pre-conditions and discrete constraints or outcomes. They are not easily extended to such weak or partial information as is provided by structural and lexical frequencies, prosody, semantics, pragmatics, and context, and doing so certainly cuts down on clarity and predictability. Yet, these seem to be the factors that enable us to resolve most lexical and structural ambiguities in natural sentences with little apparent difficulty. Strong garden-path effects are usually only achieved when these factors are aligned against the correct interpretation. Because most symbolic models do not rely on such non-syntactic sources of information, they do not provide very comprehensive accounts of human behavior in interpreting ambiguous sentences.

Although symbolic models are relatively easy to design, it is difficult to incorporate learning into them. If provided with a good teacher, such a model can quite easily learn rules. The teacher states or provides clear evidence of the rule which the model then incorporates into its processes and can potentially use to generalize to novel situations. Unfortunately, the world does not provide us with good teachers of language. As I stated earlier, parsing is not normally taught, and some have argued that neither are comprehension and production. Children receive little explicit feedback when they make a mistake and, when they do get feedback, are generally not very receptive to it (Pinker, 1984; Morgan, Bonamo, & Travis, 1995). Because symbolic models typically rely on clear feedback to support learning, the apparent lack of it has led many theorists to believe that much of our knowledge of language could not have been learned and must, therefore, be innate (Berwick, 1985; Morgan & Travis, 1989; Marcus, 1993).

[4] As I discuss the properties of symbolic models, there is always the risk of stating an overgeneralization that is not applicable to all models in this class. I accept that there are always exceptions, including the possibility of hybrid symbolic and connectionist systems; however, the following observations reflect what I see as characteristics of prototypical symbolic models.

An alternative possibility is that negative evidence is available in another form—in the frequency distribution of words, structures, and sentences in the language. Although we are not usually told explicitly that a particular structure is not part of the language, if we wait long enough and observe that it is never used by others and that alternative structures appear in its place, a reasonable learner might eventually conclude that the structure is not in the language. But to reach such a conclusion rationally, we must have a guarantee that observed usage frequencies in a language will be representative of future frequencies. Because this seems to be a valid assumption for natural language, statistical information can provide a form of implicit negative evidence that, in theory, can drive learning (Horning, 1969; Angluin, 1988).

But learning based on sensitivity to statistics creates a whole new set of problems for the designer of a model. Specifically, it raises the question of which statistics should be measured. If the model doesn't keep track of statistical information in some form, it cannot use that information. One could simply memorize every sentence one had ever heard and then extract the appropriate frequencies once they are needed. But this would be computationally intractable and not a very plausible model of the human language system. Should we just remember the frequency of words and identifiable structures? How about the frequencies of particular words in particular structures? Or the frequencies of pairs of words playing particular roles in the structure? What about the frequency of the structure in various contexts? Which contexts should be considered? One who wishes to incorporate sensitivity to statistical information into a symbolic model must decide how to answer all such questions. Bounds must be placed on exactly what information from the environment the model should keep track of. There are no easy answers to these questions and no specific bounds have, to my knowledge, been empirically justified. For this reason, frequency sensitivity and learning have not been a feature of most symbolic sentence processing models, and the idea that most of our knowledge of language is not learned but innate has become a predominant view. I have discussed these issues of language learnability, innateness, and implicit negative evidence at length in earlier works (Rohde & Plaut, 1999, in press), and will not say much more about them here.

In summary, symbolic models are useful for their high degree of explanatory clarity, but they are not well suited to account for many aspects of human language processing that seem to come naturally and effortlessly to us. These include representing a large space of overlapping and graded semantic concepts, making use of multiple sources of weak constraints to help resolve syntactic ambiguities, and, I will speculate, learning from implicit negative evidence provided by statistical information in the environment. In contrast, symbolic models are very well suited to representing abstract syntactic constructs, such as arbitrarily nested relative clauses. But syntactic complexity is the one aspect of language with which we have the most apparent difficulty. Thus, our human strengths seem to be the weaknesses of symbolic models, and their strength to be our greatest weakness.

When researchers do develop symbolic models of human comprehension performance, arbitrary limits must be placed on the models to prevent them from being too good at handling nested structures. Limits such as a maximum of three incomplete syntactic dependencies are necessary to capture the empirical data but have little or no theoretical justification. One could argue that they reflect something like working memory limitations. But one should then ask why such limitations on memory would exist. If the human language processing faculty is essentially symbolic in its architecture, shouldn't it be possible to raise these limits by just committing some more resources—some more of the billions of available neurons—to working memory? Certainly there is a selective advantage to increased working memory. If memory is a commodity that can be easily expanded, as in most symbolic architectures, why has evolution settled for such a small bound on our working memory capacity for language and for other cognitive tasks?

I propose that this is not an oversight of evolution, but that our language systems are, in fact, highly optimized. Our limitations on working memory and syntactic processing are not arbitrary. They represent the best that evolution could do with the available architecture. This is an architecture that can make use of weak statistical evidence and graded constraints, and one that can store vast numbers of lexical/semantic associations, but for which working memory does not come easily. It is an architecture that is inherently non-symbolic.

1.4 Properties of connectionist models

Currently, the principal alternative to symbolic models of cognition are connectionist networks. Connectionist networks all share the property that they process information through the interaction of simple computational units which are, at an abstract level, somewhat analogous to neurons. These units experience varying activation states and are connected with one another through weighted links over which activation flows. Units are typically arranged in layers, with all units in one layer sending connections to all units in the next layer. If the weight on a link is positive, the sending unit will tend to excite, or increase the activation of, the recipient. If the weight is negative, the sender will inhibit the recipient. This effect becomes stronger with an increase in either the activation of the sending unit or the magnitude of the link weight. The activation state of a unit is based on the net effect of the excitation and inhibition produced by its incoming links.

Localist neural networks have some similarities to symbolic models, in that the activation of each individual unit stands for some articulable property of the world. For example, the activation of unit 37 might represent the chance it is going to rain today. But in *distributed* connectionist models, each unit does not necessarily have a clear significance and information is represented by the pattern of activation over a collection of units. Information processing occurs as activation flows through the network, and is determined by the response properties of the units, their connectivity pattern, and the weights of the connections. Information need not flow in just one direction through a network. *Recurrent* networks are those in which the connections form cycles allowing information to flow around and around in the network and for new information to be integrated with old information. This enables recurrent networks to address tasks that involve processing of sequential information, such as language comprehension and production.

The link weights on distributed networks are rarely set by hand. The proper weightings to solve a given task are learned through exposure to a series of training items that specify both the inputs to the network and its correct outputs. In a digit recognition task, the inputs might be images of the digits, encoded as unit activations, and the output might be an encoding of the numerical value of the digit. In a sentence comprehension task using a recurrent network, the input might be a sequence of words and the output a representation of the meaning of the sentence. A number of learning methods are possible in connectionist models, but the most commonly used are variants of the *backpropagation* algorithm (Rumelhart, Durbin, Golden, & Chauvin, 1995; Rumelhart, Hinton, & Williams, 1986), in which error is assessed based on the mismatch between the actual outputs of the network and its desired outputs and link weights are updated so as to try to reduce this error in the future.

In addition to the fact that they are inspired by actual networks of neurons in the brain, connectionist networks have a variety of properties that make this a promising architecture in which to model the human language faculty. To begin with, distributed representations are well-suited for capturing the gradedness of semantics. One can think of a pattern of activations over a set of units as a point in a high dimensional semantic space representing a meaning or concept. Although the dimensions themselves are discrete in a representational space such as this, the positions of the concepts along the dimensions, which relate to unit activations, need not be. Complex similarity relationships among concepts can be encoded based on the proximity of their representations along the different dimensions. Although such representations can be hard for us to picture or work with, distributed connectionist networks are quite good at using representations of this form. One of the earliest and best studied uses of connectionist networks is as an associative memory that can store and retrieve large numbers of arbitrary associations (Hinton & Anderson, 1981; Kohonen, 1984). This is exactly what is required for learning and using the arbitrary mappings between word forms and the concepts they refer to in natural language.

Another advantage of using connectionist networks for language processing is that, because the units base their activations on information from all of their incoming connections, the units, and hence the network as a whole, are naturally able to rely on multiple sources of weak constraints as they process information. Integrating information from many sources is just what the units do. Therefore, as long as information from context, prosody, and semantic and pragmatic plausibility is available, the model should be capable of learning to use this information to guide its sentence processing.

One complaint often leveled against connectionist networks is that they are only able to form stimulus-response associations. Therefore, they should have been discarded as a serious account of human language processing along with behaviorism. While this may be true of the simplest one-layer networks,[5] it is not true of multi-layer backpropagation networks composed of units with non-linear response properties, which is usually what is meant by connectionist networks these days. Multi-layer networks, during training, have the ability to develop higher-level internal representations that represent abstract properties of the environment to which the network is exposed.

For example, Elman (1990) trained a multi-layer, recurrent network to predict words in simple sentences. In the input and output to the network, the words were all encoded with localist representations, with one bit for each word.

[5] When we refer to a one-layer network, we usually mean one layer of links, which actually means two layers of units.

Thus, the model was given no explicit information about word class or meaning. But after it had been trained, the hidden layer, or internal, representations developed by the model were quite interesting. In a clustering analysis, the representations for verbs were well separated from those for nouns. Within the nouns, the animates were distinct from the inanimates and the human animates distinct from the animals. Thus, just based on the statistics of the environment, the model seemed to have developed a working sense of such abstractions as word class, animacy, and humanness.

Their ability to develop higher-level representations or abstractions is what enables connectionist networks to exhibit non-trivial generalization. If there is a high degree of similarity in the internal representations of the verbs, for instance, weight updates that affect one verb are likely to affect other verbs in a similar way because their representations engage some of the same pathways. When the model learns something new about one or more verbs, that knowledge is likely to transfer, or generalize, to the rest of the verbs. Thus, generalization is possible in a connectionist network, but it operates in a somewhat different fashion than it would in a symbolic model, where it typically results from shared category membership specified by the designer of the model, rather than from shared representational properties that may be the result of learning. The ability to develop abstract representations is also one of the principal factors that sets modern connectionist networks apart from the earlier program of associationist behaviorism, which assumed that only observable stimuli, responses, and reinforcements play a causal role in behavior and learning.

The fact that connectionist networks are sensitive to the statistics of their environment is another critical property for language processing. As discussed earlier, not only is sensitivity to frequency apparent in the language processing behavior of human subjects, it is potentially able to get us past the learnability problem that results from the scarcity of explicit negative evidence in the learner's environment. In fact, a complaint I have heard several times about connectionist networks is that they are "merely sensitive to frequency." Let's consider what it could possibly mean to be *merely* sensitive to frequency.

As discussed in the previous section, there are many levels on which a model could use frequency. At the most trivial level, it could try to memorize all sentences it had been exposed to. This would require vast resources, but would be of little use unless those same exact sentences were encountered repeatedly. Any reasonable use of statistical information in language processing requires generalization. If we are to track the frequencies of words, we must be able to recognize the equivalence of words in different sentence contexts spoken by different speakers. If we are to track the frequencies of word classes or syntactic structures, we must have higher-order representations of those categories. Assuming that these are available, the question becomes exactly which frequencies to keep track of. Will we record the overall frequency of sentential complements? The frequency of sentential complements given a particular verb form? Given just the verb root, abstracting over tense? Or perhaps the frequencies should be based on the individual sense of the verb? Should we also take into account the subject of the verb? After all, we might say that a person or a dog *heard Sally*. But it would be strange for a dog to *hear that Sally is pregnant*. Thus, the frequency of a verb using different argument structures may depend on properties of its subject.

Being sensitive to statistics in the environment seemingly requires that we are able to formulate abstractions and make countless choices about exactly which frequencies to pay attention to. There is certainly nothing mere about it. To my knowledge, this problem has not been seriously addressed in any symbolic models. One solution has been to claim that frequency is not interesting and can thus be ignored, leading to the learnability problem. Another solution is to try to specify in advance exactly which frequencies the model will be sensitive to, leading to a dilemma. If we keep track of frequencies that are irrelevant to the task at hand, we will be wasting resources and we may be distracted by inconsequential distinctions and thus less able to generalize effectively. But if we fail to keep track of a useful measure, we will never be able to take advantage of it to better understand the environment or perform the task. This will be an insurmountable barrier to our learning ability.

What we need is a system that can adaptively find the most useful levels of analysis to solve a given task. This is done in a limited way in some decision trees, which are a common symbolic machine learning tool. Large trees are constructed during training, which are then pared down, based on statistical measures, to simpler ones. This is akin to paying attention to many different frequencies early on and then discarding the ones that do not seem useful. However, decision trees are quite limited in the abstractions they can form and thus the types of statistics they can be sensitive to.

In my experience, connectionist networks seem to be quite good at finding the appropriate level of analysis for a given problem. Early in training, the network can learn only about obvious properties of the environment. For example, Elman's network might have begun by learning that there is a distinction between nouns and verbs because certain words occur at the beginning of the sentence and others occur next. Then the model would be able to progress to more detailed properties, such as the fact that certain nouns cluster together in the types of verbs that follow them, and that there is a difference between animate and inanimate nouns. Later, the model may build on this knowledge when it discovers that animate nouns can use the verb *move* transitively or intransitively, but inanimate nouns can only use it intransitively. The model is under constant pressure to perform the task, word prediction, as well as it possibly can. If the model begins to pick up on a useful statistical fact about the environment, that fact will be reinforced and the model will pay more attention to it and refine it in the future. It is likely that the model will also begin to pick up on other distinctions that are not useful. Since incorrect or irrelevant information will not be reinforced, the model, under pressure to perform the best it can with its limited resources, will eventually devote those resources to tackling other problems.

The processes of deriving higher-order representations and learning from frequency information work hand-in-hand in connectionist networks. Higher-order representations develop from statistical similarity of function among lower-order representations, and the statistics that can be monitored and used by the model depend on the available representations. Connectionist models will fail to learn only when they are unable to build the appropriate representational scaffolding to discover the necessary statistical regularity in the environment.[6]

This leads to the next issue, which is whether connectionist networks are able to deal with embedded structure and other syntactic complexities in natural language, or whether that will be beyond their scope. A decade ago, many believed that representing embedded structure would be among the capabilities that recurrent networks would be unable to develop. However, a number of studies have since shown that networks can handle sentences involving limited embedded structure or cross-dependencies (Elman, 1991; Giles, Horne, & Lin, 1995; Rohde & Plaut, 1999; Christiansen & Chater, 1999b).

An interesting fact about recurrent neural networks is that working memory is not simply a commodity for them. A recurrent network that is trained from a random initial state must first learn that it has a memory, meaning an ability to make use of information from a previous state, and then it must learn how to use that memory. Working memories, to the network, are just representations of information. If the information is to be retained for long periods of time, it must be reinforced and re-represented. One cannot simply add memory to a working network to improve its performance as we can with a computer. Thus, although networks can make use of working memory to solve sequential tasks, memory of this form does not come cheaply.

In summary, connectionist networks seem to possess a number of properties that recommend theirs as a viable architecture for modeling human language. Distributed representations are quite natural for expressing the gradedness of semantic space; and, like humans, networks are well suited for memorizing large collections of associations, such as those between words and meanings. Connectionist networks are also able to integrate multiple sources of weak constraints in solving a problem, to develop higher-order representations that support generalization, and to learn from statistical information extracted from the environment—which all appear to be critical features of the human language system. While it is true that networks have a limited ability to develop working memories and hence to process deeply embedded structure, it seems that humans share these limitations. Thus, the apparent strengths and weaknesses of the human language system align much better with those of connectionist models than they do with those of symbolic models. This is why I believe that our attempt to understand human language at a mechanistic level will be most successful if we adopt a connectionist perspective.

Despite the tremendous promise of connectionist networks as models of human language processing, the reality has been rather humbling. As I will discuss in Chapter 2, all or most connectionist language models to this point have been very limited in their coverage. There has been quite a bit of study of prediction networks, based on the work of Elman (1990, 1991), alluded to earlier. These networks have demonstrated an ability to represent complex sentence structure. But prediction is, at best, a small part of language processing and does not get at the mapping between surface forms and messages which lies at its heart. There have been several connectionist models of comprehension and production, but these were, essentially, limited to working with simple sentences. The problem of representing the meanings of complex, multiclausal sentences is a difficult one which precludes the easy extension of these models. Furthermore, I am aware of no connectionist models of sentence processing that address the relationship between comprehension and production.

[6]This is, at least, my understanding of learning in connectionist networks, and I think other connectionists would agree with this analysis. However, many of these claims have not been clearly verified experimentally.

1.5 The CSCP model

The project reported in this thesis synthesizes and extends much of this earlier connectionist modeling work into a broad account of human language processing, realized in the Connectionist Sentence Comprehension and Production (CSCP) model. This model has been trained to comprehend and produce sentences in an artificial language, described in Chapter 5, which is a relatively complex subset of English. The language includes a significantly larger vocabulary than was used in previous connectionist models, as well as important features such as relative clauses, prepositional phrases, sentential complements, determiners, adjectives and adverbs, multiple verb tenses, passive voice, ditransitives, subordinate and coordinate clauses, and lexical ambiguity.

The CSCP model, which is explained in detail in Chapter 6, addresses the problem of how the meanings of complex sentences can be represented in an extendable manner and yet used effectively for comprehension and production. It learns to comprehend sentences that arrive as sequences of phonologically encoded word forms and exhibits successful comprehension by its ability to answer questions about the content of the derived message. The model also learns to produce sentences given a desired message and provides an account of the relationship between comprehension and production, suggesting that these are highly integrated systems. A major claim inherent in the model's design is that learning language production is primarily based on the formulation and testing of covert predictions during comprehension.

One of the primary goals of this thesis is to demonstrate that connectionist networks are capable of comprehending and producing complex sentences in a language with a moderately large vocabulary. As shown in Chapters 7–12, the model is indeed able to perform quite well on many types of complex sentences. It also seems to have no more difficulty in comprehending novel sentences than it does sentences on which it had been trained, thus exhibiting one important form of generalization. Furthermore, the model appears to be quite capable of handling lexical ambiguity, both across sense and across word class. While general findings such as these are interesting, if any model is to be taken seriously as an account of human language processing, it must be able to replicate human behavior in specific experimental conditions. Therefore, the model was tested in replications of psycholinguistic experiments across a range of domains, including:

- The relative difficulty of comprehending a variety of sentence types.
- Comprehension of ambiguous high versus low adverb attachments.
- The effects of reduction, past-participle ambiguity, and subject plausibility on the main verb/reduced relative ambiguity.
- The effects of reduction, verb-bias, object plausibility, and ambiguous NP length on the sentential complement ambiguity.
- The effects of verb transitivity and object plausibility on the subordinate clause, or NP/0, ambiguity.
- Grammaticality judgments on a range of variations of the NP/0 ambiguity.
- The effectiveness of semantic recovery on the NP/0 ambiguity.
- Comprehensibility of single relative clause sentences as a function of clause location and extraction type.
- The effects of prime type and target delay on structural priming.
- Agreement attraction errors in production.

One may well ask what is the usefulness in attempting a model with such broad coverage if there are likely to remain areas in which it falls short. A more usual approach to modeling cognitive phenomena is to focus on a specific domain and try to closely match all available experimental results in that domain. While, ultimately, it should indeed be our goal to explain all available data, there are several potential pitfalls in starting with too narrow a focus. The worst of these is the possibility of overfitting the phenomena. It is useful to think of a model, abstractly, as a collection of mechanisms. The amount of data we have in cognitive psychology is limited, but the space of possible mechanisms is tremendous. If a model is designed with the goal that it perform in a certain way on a small number of tasks, it is tempting to simply build into the model the most obvious mechanism for solving each task. A model designed in this way, whose mechanisms must be changed to account for each new piece of data, is unlikely to generalize well to any new tasks and is unlikely to provide the proper account of the tasks for which it was designed.

Overcoming the constraints of modeling data from widely ranging areas is what will ultimately enable us to make progress in developing models of language processing or other cognitive phenomena. It is doubtful that a simple model will perform well on a range of tasks unless it is truly on the right track. The most obvious or easily implemented mechanism for explaining a particular behavior may not be the correct one. But a mechanism that can explain data from many different experiments is much more likely to be close to the truth. Consider a situation in which mechanism X is an obvious solution to problem A, but also provides a non-obvious, albeit correct, solution to problems B and C. If we have tried to explain domain B by focusing solely on it, we may never discover the correct mechanism. Building models that account for a broad range of phenomena, rather than those in a focused domain, is much more challenging, but if you can pull it off the rewards are far greater.

Because the CSCP model is still in the early stages of development and because its scope is quite broad, when evaluating the model I am primarily interested in its ability to account for qualitative patterns of human behavior. It is important to keep in mind that the model's parameters were not adjusted to achieve a particular pattern of performance on any one experiment. The details of most of the experiments conducted on the model were not even planned at the time it was designed. The final version of the model was trained once—a two-month ordeal—and then tested extensively, without further training. The model was designed primarily with the goal of achieving the best possible overall performance with the given computational resources, and the statistical properties of the language on which it was trained were based as closely as possible on those of English.

Although it is certainly not perfect, and I will try to let the results speak for themselves, I think it is fair to say that the CSCP model was able to replicate many key aspects of human behavior in the areas on which it was tested. The model is able to generalize from its training, to resolve temporary ambiguities, and to be appropriately sensitive to lexical and structural frequencies, semantic plausibility, and locality effects. The majority of cases in which the model does not perform as expected seem to be traceable to artificial computational limitations placed on it and to the poor design of some of its interfaces, rather than to fundamental flaws of its connectionist architecture. I believe this work represents a major advance in the attested ability of connectionist networks to process natural language and a significant step toward a more complete understanding of the human language faculty.

1.6 Chapter overview

Chapter 2 contains a review of earlier connectionist models of language processing, including models of parsing, prediction, comprehension, and production. Chapter 3 is a review of the psycholinguistic literature, covering empirical results on some of the major topics of investigation in the field, including the comprehension of sentences with relative clauses and structural ambiguities as well as sentence production. Many, but not all, of the findings reviewed here are addressed by the model in later chapters. Because the CSCP model is so sensitive to the statistics of its environment, the language on which it is trained must be statistically representative of English if the model is to provide a valid account of the performance of English speakers. Therefore, designing this language, which is known as Penglish, begins with a detailed analysis of the statistics of English, which is reported in Chapter 4.

Chapter 5 describes the actual Penglish language itself, including its syntax, how its semantics are represented, and the phonological encodings of its words. Finally, in Chapter 6, we get to a description of the CSCP model. This covers the model's structure, the way in which it encodes and decodes sentence meanings, how it performs comprehension and production, and how it is trained and tested. This chapter also describes how reading times and grammaticality measures are extracted from the model, and discusses some of the rationale behind its design and the recognized limitations that have been imposed on it.

The next six chapters contain analyses of the model's performance. Chapter 7 involves general experiments on the model's comprehension ability, covering such issues as the difficulty of various sentence types, generalization, lexical ambiguity, properties of the model's reading time measure, and individual differences between networks. Chapters 8, 9, and 10 focus on the main verb/reduced relative (MVRR) ambiguity, the sentential complement (NP/S) ambiguity, and the subordinate clause (NP/0) ambiguity, respectively. Each summarizes the relevant empirical results and then reports analogous experiments conducted on the model. Chapter 11 considers the model's ability to process sentences with various types of relative clauses. Chapter 12 is devoted to production and reports general results, contains a detailed analysis of production errors on several sentence types, and includes experiments on agreement attraction errors and structural priming.

Finally, Chapter 13 is a summary and discussion of the model's performance, properties, and relationship to other theories of human language processing. There are four appendices, the first three of which describe software tools that I have developed in the course of my graduate studies to make this project possible. Appendix A is an introduction to the LENS neural network simulator, which is of general usefulness to connectionist modelers, but which was customized to implement the CSCP model. Appendix B describes a program called the Simple Language Generator (SLG), which was used to produce and recognize the Penglish language. It generalizes stochastic context-free grammars to allow the integration of semantic constraints. The third tool, described in Appendix C, is TGREP2, which is a reimplementation and extension of the TGREP program for analyzing syntactically parsed corpora. Finally, Appendix D contains some more detailed specifications of the Penglish language.

Chapter 2

An Overview of Connectionist Sentence Processing

The CSCP model introduced in this thesis builds on a wealth of earlier research into connectionist approaches to sentence processing. In order to place the current work in its appropriate context, this chapter reviews the most significant, or at least well known, connectionist sentence processing models to date. There have been a number of similar compilations in the past (Diederich, 1989; Sharkey & Reilly, 1992; Hahn & Adriaens, 1994; Wermter, Riloff, & Scheler, 1996; Christiansen & Chater, 1999a; Steedman, 1999), but we will focus on models that address the semantic and syntactic issues involved in handling multi-word utterances and will ignore most of the important applications of connectionist networks to other phenomena such as word reading, lexical decision, and past tense formation.

The models discussed here reflect a general progression from localist implementations of symbolic systems to systems of interacting localist units to distributed representations and multi-layer learning rules and finally to recurrent networks capable of learning. Although the early localist models are discussed, most of the later localist or hybrid symbolic/connectionist systems have been excluded since they typically differ from symbolic systems only in the implementational details. If the reader is interested, a number of hybrid systems are reviewed in Wermter et al. (1996).

Most sentence processing models are designed to address one of four major language tasks: parsing, comprehension, word prediction, or production. The models are grouped by the primary task for which they were designed, rather than in chronological order.

2.1 Parsing

Parsing, or producing a syntactic, structural description of a sentence from its surface form, is the one sentence processing task that has received the most attention from the symbolic community. Indeed, given that the traditional approach to language has minimized attention to semantics, parsing is one of the few behaviors left that, ostensibly, may not rely on semantics.[1] Thus, it should not be surprising that many of the connectionist parsing systems found in the literature are essentially symbolic models implemented transparently in connectionist hardware.

Learning has not played a major role in most of these parsing models for two main reasons. First, most connectionist parsing models have been localist. This architecture lends itself to hand-designed weight structures but not to the easy design of effective learning environments. But more critically, teaching a model to produce an explicit parse of a sentence requires, for most systems, training data labeled with correct parsing information. Few believe that such information is actually available to the child, so models which rely on it are of questionable relevance to human learning.

I would argue that this issue points to a fundamental problem with the view that parsing in and of itself is a core

[1] The extent to which there exists an independent syntactic parsing module has been a matter of considerable debate, but there now seems to be widespread skepticism over the presence of even a first-stage parser that is purely syntactic (McClelland, St. John, & Taraban, 1989; Seidenberg & MacDonald, 1999).

component of human language processing. We have clear evidence that humans are able to comprehend and produce language. However, that humans actually engage in explicit parsing during comprehension is an assumption that could reasonably be questioned. There seems to be little direct evidence that we construct a representation of a syntactic parse tree while comprehending. Thus, although parsing may be useful in computational linguistics, if one is really interested in understanding the human sentence processing mechanism, parsing may not be the appropriate level at which to focus. Perhaps for these reasons, the early interest in connectionist parsing mechanisms seems to have waned in recent years.

2.1.1 Localist parsing models

The first significant proposal for a connectionist model of parsing, **Small, Cottrell, and Shastri (1982)**, does not actually fit the pattern of a transparently symbolic approach. Following McClelland and Rumelhart (1981), this model stresses the importance of interaction between syntactic information and semantics, or general memory skills. This interactive activation approach contrasts with more standard parsing theories that stress compartmentalism and serial processing (Frazier, 1979; Fodor, 1983). The Small et al. model, implemented in later work **(Cottrell, 1985b)**, is not actually a full parser but is designed for word-sense disambiguation, which is arguably an important sub-task of parsing and comprehension. The model uses localist units to represent lexical items, individual word senses, and case roles. These units excite or inhibit one another through a set of hand-designed connections. Because of this, the model is not easily expandable to larger vocabularies or complex linguistic structures.

Cottrell (1985a) extended the earlier work with the addition of a full-fledged syntactic parsing network. The network can be generated automatically given a grammar, but still requires some weight-tweaking. Concepts are associated with case roles by means of localist *binder* units. There is a unit for each concept/role pair and these units mutually inhibit one another. Units in the syntactic portion of the network represent the non-terminal symbols of the context-free grammar, and their interconnections reflect the possible productions in the grammar.

The model is interesting in that it is able to process sentences presented in a temporal sequence and makes use of interacting top-down and bottom-up information. However, it has a number of limitations. As is a common problem with other models that make use of case-roles, the model does not appear capable of handling sentences with multiple verbs. It can also handle only fixed-length sentences and requires constituent recognizers with duplicated and possibly non-connectionist control structures. Finally, some might complain that the model is not guaranteed to settle into a single, coherent interpretation of any sentence.

Several other connectionist parsing models appeared at about the same time. Except where noted, they are localist, non-learning models. Because they also use a fixed-size network and a static input representation, rather than temporally coded inputs, these networks are able to process sentences only up to a finite length and often rely on redundant structure, as in the Cottrell (1985a) model. The main problem with redundant structure is its lack of parsimony. For example, a typical parsing system involving redundant structure might have a separate NP-recognizer for every context in which a noun-phrase might occur, rather than a single recognizer that operates in all contexts. Along with an inefficient use of resources, this creates a problem for learning. Does each of these recognizers undergo learning only when it is used, resulting in a lack of experience for those that are used rarely, or are they all tied together in some way? Either answer seems to introduce a variety of logistical problems.

Waltz and Pollack (1985) presented an interactive activation model which differs from other work in that it does not consist of a single network. Rather, the network is generated based on the grammar and the sentence. This network is able to represent only the possible parses of the given sentence. A settling phase allows the network to settle into a particular interpretation. The model has a number of drawbacks, most significant of which is that it is not grammar-general but uses a program to produce a specific network for each sentence. The networks also do not process words over time but use a static input representation and thus are not able to produce partial, online interpretations of sentences. Although the implemented model was purely localist, Waltz and Pollack proposed that concepts should not be represented by single nodes but by distributed patterns of "microfeatures," a suggestion that would be adopted in later connectionist modeling.

Fanty (1985, 1994) took a rather different approach. Aiming to produce a network that is deterministic, fast, and guaranteed to work, Fanty devised a way to implement the CYK dynamic-programming, context-free parsing algorithm (Younger, 1967) in a localist network. The network is able to handle sentences up to a fixed length. It essentially contains a unit for every non-terminal paired with each sub-sequence of the input. The network operates in two passes: a bottom-up phase and a top-down phase. In the bottom-up phase, units for increasingly longer sub-sequences become active if their non-terminal could have produced the words in that sub-sequence. In the top-down phase, units that do not fit within a coherent parse are silenced. In the end, only units that participate in a legal parse remain active. Thus, the model does not require an extended period of relaxation. This model is interesting because it suggests that language may be parsable by a non-recursive procedure.

However, many natural language sentences have multiple possible syntactic parses, and Fanty's basic model is not able to select the appropriate one. Fanty considered one way to bias the model toward selecting shallower parses, but did not attempt to integrate the critical semantic information as in other models. The other major limitations of this model are that it can handle only fixed length sentences and that it relies on redundant structure. Although the model is not able to learn entire grammars, Fanty discussed how small errors in the model could be corrected through learning. **Rager (1992)** described a localist model based on Fanty's but designed to handle "extragrammatical," or slightly incorrect, sentences.

Selman and Hirst (1985, 1994) presented a model that differs from other early connectionist parsers in that it uses a variation on the Boltzmann machine (Fahlman, Hinton, & Sejnowski, 1983) with non-deterministic units and a simulated annealing scheme to allow the network to settle gradually into a stable configuration. The rules of a context-free grammar are implemented in the network by means of syntactic binder units that inhibit one another and excite other units representing symbols that participate together in a production. The use of simulated annealing, while very slow, allows the network to settle into the correct parse with high probability. However, as in other localist models, this model requires sentences to be bounded in length and uses redundant structure. Due to the proliferation of binder units, the size of the network may grow intractably with more complex grammars. Furthermore, although the authors suggested it as a next step, this model does not incorporate semantic information and it is not clear how it would deal with syntactic ambiguity.

Charniak and Santos (1987) described another localist parsing model that differs from the others in its use of a sliding input window. This allows the network theoretically to handle sentences of unbounded length but hinders the ability of the model to process long-distance dependencies, such as those surrounding center-embeddings. Although the model was successfully implemented for a very simple grammar, it is not clear that its parsing heuristics would be sufficient to handle more complex grammars. The model also uses parts of speech rather than lexical inputs and was thus clearly unable to incorporate semantics or resolve syntactic ambiguities.

Howells (1988) described an interactive activation parser known as VITAL. As in Waltz and Pollack (1985), Howells' networks were generated during parsing. However, it appears that the model could have been implemented as a single network. Given that the connection weights and unit thresholds in the model were carefully balanced, it is unclear how well it could be scaled to handle more complex languages. One interesting aspect of the model is that it makes use of the frequency with which productions in the grammar appear in sentences it has experienced. Thus, parsing can be biased toward more common interpretations and allows for a limited degree of learning. However, the model does not incorporate semantic information.

The model of **Nakagawa and Mori (1988)** also involves constructing the network on-the-fly. But a network is not built for the entire sentence prior to parsing, it is generated sequentially, essentially implementing a left-corner parser. Sigma-pi units—those with multiplicative rather than additive inputs—are used to enforce ordering constraints on grammatical structures. Although the model can theoretically parse unbounded sentences, the copying mechanism used to construct the parse tree is not physiologically reasonable. The model also does not incorporate learning or semantic constraints.

The commonalities evident in these early connectionist parsing models lead to some generalizations about the limitations of the localist approach. With localist units, where each unit represents an explicit state and only a small subset of units can be active at one time, the representational capacity of the network is proportional to its size. This leads inevitably to the problem that a fixed-size network can only handle inputs of bounded length or complexity. Like the limitations of symbolic models and unlike those of networks that use compositional distributed representations, this results in hard limits on the processing ability of localist networks. Localist and symbolic models generally do not exhibit a gradual degradation of performance with more difficult inputs, and modeling of performance data generally requires ad hoc limitations.

Learning is difficult in localist networks largely because of the problem of designing supervised training environments. This is compounded by the fact that large localist networks tend to require redundant structure, and effective learning mechanisms ought to generalize what is learned across duplicated sub-networks. It seems difficult or impossible to accomplish this in a reasonable manner. Finally, hand-wired networks do not allow easy incorporation of semantic information, even though this incorporation is necessary for parsing structurally ambiguous sentences, as aptly demonstrated in McClelland et al. (1989). Aside from the ability to incorporate multiple sources of weak constraints, localist networks provide few advantages over symbolic models.

2.1.2 Hybrid and distributed parsing models

While the last decade has seen quite a few hybrid connectionist/symbolist parsing models, only a few will be mentioned here. The CDP model of **Kwasny and Faisal (1990)** is a modification of the PARSIFAL deterministic parser (Marcus, 1980). Several of the components of this rule-based parser were removed and replaced with a connectionist network. This network was trained to suggest actions to be taken by the symbolic components of the model based on the contents of the symbol stack and the constituent inputs. The model was reportedly able to process ungrammatical and lexically ambiguous sentences in an appropriate way. However, it is not clear what effect the network component really had on the model. The primary reliance on a symbolic parsing mechanism is something most connectionist researchers would hope to avoid. The authors did recognize the need for more fully connectionist parsers and discussed some of the hurdles involved.

Stevenson's more recent parsing model **(Stevenson, 1994; Stevenson & Merlo, 1997)** is largely symbolic, but relies on activation-based competition mechanisms, as in localist network models, to resolve structural ambiguities. The model contains a set of subtree templates representing $\overline{X}$ structures. These structures contain attachment points at which they can be connected together to form complete parse trees. The trees are constructed from left to right. In cases where multiple attachment sites are possible, an activation-based competition ensues, which is influenced by factors such as recency, frequency, and context. Limited reanalysis is possible, but any attachments that are not on the right-most fringe of the tree cannot be revised. There is also a competition among the possible $\overline{X}$ structures that can project from a word, but this competition was not actually implemented in the simulated parser and it is therefore not able to handle lexical ambiguity. This model is also insensitive to purely lexical effects that do not involve argument structure differences. Because the details of the competition and structure projection aspects of this model are so complex, and because of the limits to reanalysis, it seems likely that the model will fail to parse many complex sentences that are routine for humans and that it will be insensitive to many factors that influence human comprehension.

Most grammar-based parsers suffer from an inability to handle sentences that fall outside of the given grammar. This can be a serious problem given the prevalence of pauses, false-starts, corrections, and word-substitutions in spoken language. **Wermter and Weber (1994, 1997)** and **Weber and Wermter (1996)** were interested in designing a system that was robust in the face of such problems. Their SCREEN model is a complex, highly modular, hybrid connectionist/symbolic system. While some of the modules are implemented in a symbolic manner, most are networks trained to perform a particular operation. Rather than producing full parse trees, the SCREEN model generates a *flat* syntactic and semantic parse. That is, the model labels the constituents by their syntactic class (e.g. noun or verb), their more abstract syntactic level (e.g. noun group or verb group), and some of their semantic properties including a few thematic roles (e.g. agent, action, or animate). The model was trained and tested on spontaneous spoken utterances and appears to work quite well. While the overall modular structure of the network is a symbolic design, the use of trainable, distributed networks allows for a certain level of generalization and fault tolerance. However, a serious limitation of the model, for many applications, is that the flat parse lacks much of the information necessary to construct a full parse tree. For example, the model does not appear to be capable of representing multiple interpretations of a prepositional phrase attachment ambiguity.

The **Jain and Waibel (1990)** model is essentially a localist, slot-based network, but it does incorporate learning and distributed representations at the word level. It consists of a series of layers which essentially represent words, phrases, clauses, and inter-clausal relationships. These layers are trained independently with specified targets and therefore involve only limited learned, distributed representations. The model is interesting in its ability to process inputs over time, producing expectations of sentence structure and dynamically revising hypotheses. However, it only has a fixed number of phrase and clause blocks and uses weight sharing to generalize learning across phrase blocks. This appears to cause a difficult tradeoff between proper generalization and over-generalization. It is not clear how well this model could make use of semantic information in resolving ambiguities.

Although several earlier connectionist models that were not purely parsers are described in Section 2.5, the XERIC model of **Berg (1992)** was one of the first distributed models that learns to parse. XERIC combines a simple-recurrent network (Elman, 1990) with a RAAM (Pollack, 1990) and is able to take words over time and produce a representation that can be decomposed into a parse tree whose structure is based on X-bar theory. This model has the advantage over localist methods that it can process unbounded sentences with only gradual degradation in performance. Although it was trained on a fairly simple grammar, the model is able to parse sentences with rather deep structure. Although semantic information was not included in the original work, such information could theoretically be introduced into this model by using a micro-featural encoding for words at the input. Despite its successes, XERIC might not be considered an adequate cognitive model because its hierarchical training procedure, like that for the RAAM, requires considerable memory and symbolic control. More crucial however, as with the Jain and Waibel (1990) model, is that the parsing information used to train the network is not available to the child.

Henderson (1994a, 1994b, 1996) described a localist, non-learning connectionist parser based on *temporal synchrony variable binding* (TSVB) and inspired by symbolic parsing theories. The main idea behind TSVB is that variable bindings, such as the bindings of constituents to thematic roles, can be represented by synchronous firing of constituent and role representations. The use of temporal synchrony, rather than something like binding units, reduces the need for duplicate structure and permits greater generalization. Henderson argued that the overall architecture is biologically well-motivated. The model, which is based on *structure unification grammar* (Henderson, 1990), does not itself construct an entire parse tree. Rather, it produces tree fragments with sufficient information that they could be combined into a complete tree. Because it is a deterministic parser, never backtracking on its commitments, and because it is unable to represent disjunctions of interpretations, it is likely that this model would have great difficulty with ambiguous sentences and suffer from an overly strong garden-path effect. The main drawback of the model is that it is primarily a connectionist implementation of a symbolic algorithm and lacks many of the advantages of connectionist networks, including the ability to learn and make use of multiple weak constraints.

Henderson and Lane (1998) and **Lane and Henderson (1998)** described an extension of the TSVB approach, known as a *simple synchrony network*, that can learn to parse sentences. The network takes the part of speech tags for the sentence constituents as input and is trained to produce the parse tree fragment of any constituent seen so far when that constituent is queried. Although the network never produces a full parse tree, the tree fragments could be assembled into one. The network was able to learn to parse a corpus of written English to a reasonable degree of proficiency. However, this success is bounded by the limits of relying on parts of speech rather than actual words. This model might gain some advantage from using words rather than tags as input, but it would then encounter problems of lexical ambiguity. Nevertheless, the model is rather interesting, and could potentially have reasonable practical applications. It is worth noting that TSVB seems to be identical in practice to the query mechanisms used in St. John and McClelland (1988, 1990, 1992) and in the CSCP model presented here.

Finally, **Harm, Thornton, and MacDonald (2000)** were interested in how semantic and statistical regularities affect the parsing process. To begin to address this, they focused on the parsing of phrases such as "the desert trains" which are ambiguous as either a noun phrase, as in (15a), or as an NP followed by a verb, as in (15b). Harm et al. trained a fully recurrent network (Pearlmutter, 1989) to process potentially ambiguous three-word phrases. As each word was presented, the network mapped from a distributed representation of the word's form to a distributed representation of its meaning. Also present in the output was an indication of whether the phrase was an NP or an NP followed by a verb. The network was assessed on the speed and accuracy with which it settled into the correct representation under various conditions. Although this model is quite limited, in that it is designed to handle only short phrases, it succeeded in demonstrating the desired sensitivity to a variety of factors, including structural constraints, pragmatic constraints, and lexical frequency and semantic biases.

(15a) The desert trains were late reaching the station.

(15b) The desert trains the soldiers to be tough.

2.2 Comprehension

Although parsing models have sometimes been labeled comprehension models, I use the latter term to refer to systems that aim to derive a meaning for an utterance that goes beyond its syntactic structure. There are, in fact, relatively few comprehension models in the literature. This may be due largely to the difficulty of representing and processing semantic information. Concept and phrase meanings involve subtle aspects that cannot easily be captured in a symbolic or localist system and do not interact in a cleanly combinatorial fashion. Furthermore, systems able to manipulate such information do not lend themselves to top-down design and are better constructed with learning methods. Therefore, comprehension has largely been the domain of distributed, connectionist models.

2.2.1 Comprehension of simple sentences

Hinton (1981) discussed one way in which semantic information and associations could be stored and recalled using distributed representations, and he pointed out some of the advantages this has over traditional localist semantic networks and over static distributed representations. A principal advantage is that associations formed between items may automatically generalize to semantically similar items. This work appears to have influenced, directly or indirectly, many subsequent connectionist models of semantics.

One such effort is the well-known model of **McClelland and Kawamoto (1986)**. While it does not derive fully structured representations of sentence meaning, this model produces thematic case role assignments, which are thought to be an important element of comprehension. Assigning case roles typically involves labeling the nouns in a sentence with their primary relationship to the verb heading their clause. Typical thematic roles are agents, patients, instruments, and experiencers. A key observation is that proper assignment of case roles does not simply depend on word order but also involves considerations of word meaning, inflectional morphology, and context. McClelland and Kawamoto hoped their model would be able to select the appropriate readings of ambiguous words, fill in missing arguments in incomplete sentences, and generalize its knowledge to handle novel words given their semantic properties.

The model uses stochastic units and a single layer of weights that is trained using the perceptron convergence rule. The inputs to the model consist of the semantic features of up to four main constituents of the sentence—three nouns and a verb—which are then recoded using four larger sets of units that represent conjunctions of pairs of elements from the original arrays. The model is then trained to produce the semantic representations for the fillers of up to four thematic roles: agent, patient, instrument, and modifier. The model is able to satisfy many of the authors' goals, including resolving lexical and structural ambiguities, handling shades of meaning, and generalizing to novel words.

However, as they acknowledged, this was just a first step because it greatly simplified the problem of sentence comprehension. The use of static input representations does not allow the network to process words over time and results in a hard limit on the complexity of sentences that can be handled. In particular, this model would be unable to represent multi-clause sentences without considerable changes. The elimination of function words and the use of a fixed set of output slots limit the number of thematic roles that could be recognized by the model. McClelland and Kawamoto suggested a number of ways in which these and other problems could be remedied and this was further fleshed out, though not implemented, in McClelland and St. John (1987) and McClelland (1989).

Perhaps the best known model of sentence comprehension is the later work of **St. John and McClelland (1988, 1990, 1992)** and **McClelland, St. John, and Taraban (1989)**. These papers described a model that shares many of the goals of the McClelland and Kawamoto (1986) work but extends the framework to produce a changing interpretation as each constituent is received and to allow the learning of distributed hidden representations of phrase and sentence meaning. The input half of the model is a simple-recurrent network (Elman, 1990) that learns to use a sequence of phrase components to compile a single message representation, known as the sentence gestalt, in the form of a trainable hidden layer. The phrase components are either a simple noun phrase, prepositional phrase, or verb. The output half of the model was trained to answer questions about the sentence in the form of a probe. When probed with a constituent, the network must respond with the thematic role played by that constituent. When probed with a role, the network produces the constituent that fills that role. During training, the error that derives from the answers to these probes is backpropagated through the network to influence the formation of the sentence gestalt.

The St. John and McClelland model successfully exhibited its desired behaviors, including the ability to...

- make use of both syntactic and semantic clues to sentence meaning.

- revise its interpretations online and produce expectations in the absence of complete information.
- infer missing constituents, for example, that eating soup is probably done with a spoon.
- infer properties of vague constituents, such as "person," based on context.
- handle both active and passive sentences.
- use variable verb syntactic frames.
- generalize its abilities to novel sentences.

A major limitation of the model is that it is not able to process multi-clause sentences, which are of considerable interest in the study of language. Other limitations include the representational inadequacy of a small number of fixed thematic roles and the lack of extra-sentential context. Nevertheless, the St. John and McClelland model remains a key inspiration for the work discussed in this paper.

One hindrance to the development of sentence comprehension models has been the difficulty of specifying adequate meaning representations of concepts and sentences. One solution adopted by **Allen (1988)**, **St. John (1992a)** and **Noelle and Cottrell (1995)** is to avoid specifying meanings by focusing on language learning in the service of a task. By grounding language in this way, the model can be trained to respond to linguistic inputs by performing an appropriate action. Allen (1988) described a model which takes as input a coded microworld and sequential questions about that world. The simple-recurrent network was trained to answer questions with either yes/no or single constituent responses. In similar work, St. John (1992a) trained a simple-recurrent network to take a description of a scene and a sentence identifying a particular block in the scene, such as "the big blue block that is on the right of the left page," and output the block to which the sentence refers. The model is able to handle fairly complex inputs including relative clauses and prepositional phrases and can even handle human-produced sentences moderately well, but is otherwise severely limited in its scope.

Noelle and Cottrell (1995) were interested in the ability to perform a task immediately after receiving some instructions on how to perform it, which they refer to as "learning by being told." The framework of their model was inspired by the sentence gestalt network of St. John and McClelland (1990). The *plan* component of the network receives instructions over time and produces a plan that guides the performance of the *domain task* portion of the network. In this way, the sentence gestalt model might be viewed as one in which the input sentence instructs the model how to act appropriately in the domain of answering queries about that sentence. Although Noelle and Cottrell did not phrase the instructions to their model in natural language, that would be a simple extension. The suggestion that much of language is learned in the service of various tasks is a reasonable one. However, it seems unlikely that all of language is learned through direct, action-based feedback in this way.

Miikkulainen and Dyer (1989a) trained a backpropagation network on the same sentences used in the McClelland and Kawamoto (1986) study. The network learned to map from a static representation of the words in the sentence to a representation of the case role assignments. The principal difference between this and the earlier study is that McClelland and Kawamoto hand-designed feature-based distributed representations for words while the Miikkulainen and Dyer network learned the word representations using the *FGREP-method*. In the FGREP-method, word representations are initially randomized. Error is propagated all the way back to the input units, and the word representations are updated as if they were weights on links feeding the input group. The revised representations are then used as training targets on subsequent sentences. This method seems to be an effective one in practice for learning representations when they must appear at both the input and output of a network. However, it is not clear what prevents the representations from degenerating into, for example, all zeros, nor how it could be implemented without a symbolic controller. The task performed by the system is simpler due to the fact that words maintain the same representations in the input and output. There is no distinction between phonological and semantic representations, and the meaning of a sentence is treated quite literally as the concatenation of its parts. The method was later extended to a simple-recurrent network which accepts the same sentences encoded sequentially (Miikkulainen & Dyer, 1990).

2.2.2 Comprehension of complex sentences and stories

Miikkulainen and Dyer (1989b, 1990, 1991) further extended their model to the comprehension and production of script-based stories from a limited set of domains. The stories consisted of a series of simple sentences describing activities such as eating in a restaurant or shopping. The system involves four modular networks which all share the same word representations due to the FGREP mechanism. One network maps a sequence of words into a slot-filler representation of the case roles of the sentence. The next module maps a sequence of sentence representations to a slot-filler story representation. Two other modules are trained on the inverse mappings. The networks are able to comprehend and reproduce the stories and can fill in missing details from partial stories. However, the true generalization abilities of the system are questionable given that the stories are drawn from a very restricted set of possibilities. While the use of modules improves the ability of the network to solve this task, the method relies on encoding sentences and stories with visible, slot-based representations. This does not extend easily to the more complex and subtle aspects of natural language.

Miikkulainen (1990b) applied the modular architecture to comprehending and producing sentences with relative clauses. The network is similar to that used to process stories. The sentences were composed of noun-verb or noun-verb-noun clauses, separated by commas. The first module maps from a sequence of words drawn from a single clause, or part of a clause if it contains an embedding, to a slot-based representation of the meaning. A second network maps from a sequence of clause frames to a static representation of all the frames in the sentence. Two other networks perform the inverse mappings. The system was trained on a set of 388 sentences with up to 3 clauses utilizing just 3 different verbs and 4 nouns and was able to reproduce the sentences quite well. The use of a slot-filler representation for sentence meaning places a hard constraint on the complexity of sentences that could be represented by this system. Another limitation is that it relies on markers to distinguish clause boundaries, thus preventing it from handling reduced-relative constructions, which lack relative pronouns. Nevertheless, aside from the current work, this appears to be the only connectionist comprehension model able to process complex sentences.

Two other connectionist comprehension models, **Miikkulainen (1990a)** and **St. John (1992b)**, also address the problem of comprehending stories with multiple sentences. Both use sequences of propositions encoded in thematic role frames, rather than actual sentences, as input. For example, `(agent=person1, predicate=drove, patient=vehicle, destination=airport)`. The Miikkulainen model uses self-organizing feature maps to form an unsupervised classification of stories based on the type of event being described. The St. John model, known as the *story gestalt*, is quite similar in design to the earlier sentence gestalt models (St. John & McClelland, 1990). However, it was trained to answer queries about entire stories rather than individual sentences. The main issues addressed by that model are the representation of multiple propositions, resolution of pronouns, revision of on-going interpretations and inferences, and generalization, under the hypothesis that graded constraint satisfaction plays a primary role in these processes. The model was quite successful except for weakness in its generalization abilities.

2.3 Word prediction

Some of the most successful connectionist models of sentence processing are those that perform word prediction. That is, at any point in the sentence, the model should be able to produce the probability that each word might occur next. Word prediction is a surprisingly useful ability. It can be the foundation for a *language model* which predicts the likelihood that a particular utterance will occur in the language. This is a principal component of most speech recognition systems since it is quite helpful in resolving ambiguous inputs. The ability to predict accurately is sufficient to generate the language, and it thus indicates knowledge of the grammar underlying the language. As a result, prediction networks are sometimes labeled *parsers*. However, that term is reserved here for a model that produces an explicit representation of the syntactic structure of the sentence.

2.3.1 Elman (1990, 1991, 1993) and the issue of starting small

The best known connectionist prediction models are those of **Elman (1990, 1991, 1993)**, who pioneered the use of simple-recurrent networks (SRNs), also called Elman networks.[2] Elman (1990) applied an SRN to letter prediction in a concatenated sequence of words, demonstrating that the network could potentially learn to detect word boundaries by identifying locations of high entropy, where the prediction is difficult. This work suggests that prediction might be a primary mechanism used by infants to learn word segmentation. Elman then extended the model to word prediction

[2] A simple-recurrent network (SRN) is much like a standard feedforward network that operates in discrete time steps. However, one or more layers can receive projections which receive input from the activation of a layer at the previous time step. In the traditional Elman network, the hidden layer receives one of these delayed projections from itself.

in a language of simple sentences. Representations of words that developed at the network's hidden layer could be clustered to produce a reasonable classification of words syntactically and semantically. This indicates that much of the basic knowledge required for parsing and comprehension could be extracted by a prediction mechanism from the child's input.

Elman (1991) further extended the model to process sentences that potentially involve multiple embedded clauses. The main goal of this work was to demonstrate that networks are capable of learning to represent complex, hierarchical structure. This is clearly a critical question if one is concerned with their ability to process natural language. As Elman put it, "The important result of the... work is to suggest that the sensitivity to context which is characteristic of many connectionist models, and which is built-in to the architecture of [SRNs], does not preclude the ability to capture generalizations which are at a high level of abstraction" (p. 220). A second major outcome of the work was the finding that the networks were only able to learn corpora of mostly complex sentences if they first began training on simple sentences before gradually advancing to a higher proportion of complex ones.

This was developed further in Elman (1993), where it was shown that the networks could also learn well if their memory spans were initially hindered and then gradually allowed to improve. This finding was thought to be particularly important as it accorded with Newport's "less-is-more" hypothesis: that a child's limited cognitive abilities may actually be a critical factor in enabling her to, ultimately, learn a first or second language to a greater degree of fluency than can an adult (Newport, 1990; Goldowsky & Newport, 1993).

These results appeared to have important implications for human language learning. They suggested that simple-recurrent networks, and by extension perhaps recurrent networks and the human brain, can learn to process sentences only if they are exposed to inputs of gradually increasing complexity. Elman argued that this progression could come about in two different ways. Either the environment itself could change or the learner could change. By starting with faulty or limited memory, it was argued, the learner naturally filters out complex inputs early on. As its memory improves, more complex inputs are experienced intact; and the result is, in a sense, much like the environment gradually introducing more complexity.

However, in work that was a precursor to the current project, **Rohde and Plaut (1997, 1999)** re-examined these findings and discovered that manipulating the training environment or memory span of the networks does not always facilitate learning and can, in fact, be harmful. These studies used a similar network to Elman's but a range of languages that differed in their statistical, but not syntactic, properties. The primary finding was that using initially simplified inputs was, in most cases, a significant hindrance to the networks. This was particularly true as the languages were made more natural through the introduction of semantic constraints.

Memory impairments of the sort used by Elman, on the other hand, seem to have little effect on the learning of the network. Our explanation for this was based on the fact that recurrent networks naturally begin with poor memory which they must gradually learn to use as they are exposed to the environment. The network therefore tends to learn simple relationships first because it does not yet have the representational capacity to handle more complex ones. Thus, Elman's staged memory impairments tend to have little effect because they simply mirror the natural development of memory. Memory is poor initially so frequent interferences cause few problems. As memory improves, interference occurs less often, so it continues to have little practical effect.

Those interested in the development of language should avoid thinking of short-term or working memory as an innate capacity. Our experience with neural network models suggests that working memory does not rely on a preformed memory module, as in a computer. It is the result of information sustained and transformed by the activation of neurons. The interaction of those neurons is not completely determined in advance but gradually develops through experience. Memory is not an innate capability but a learned skill.

To state that memory develops gradually in the language learner might seem to be an endorsement of Newport's less-is-more hypothesis, but there is an important distinction. There is little evidence that memory limitations in and of themselves can be beneficial to learning a complex skill like language. As argued in **Rohde and Plaut (in press)**, "We believe that the cognitive limitations of children are only advantageous for language acquisition to the extent that they are symptomatic of a system that is unorganized and inexperienced but possesses great flexibility and potential for future adaptation, growth and specialization."

One final caveat on this topic pertains to the issue of starting with simplified sentences. Although we found this manipulation to be a hindrance to the prediction network, Rohde and Plaut (1999) made the point that simplified input may prove more beneficial for the task of comprehension. Beyond simply learning the grammar, comprehension requires the learner to associate meanings with surface forms in the language. "This process is certainly facilitated by exposing the child to simple utterances with simple, well-defined meanings" (p. 98). Although starting with simple inputs may be helpful in learning comprehension, we expect there to be a tradeoff and would predict that an extended period of exposure to simplified language will result in developmental impairments.

2.3.2 Other prediction models

Having digressed somewhat, we return to the review of sentence prediction models. The remaining connectionist prediction models are all based more or less directly on Elman (1991). **Weckerly and Elman (1992)** focused specifically on the issue of the difficulty of right-branching versus center-embedded sentences. They found that, in accordance with behavioral data, the SRN showed a preference for sentences involving double right-branching, subject-extracted relative clauses, as in (16a), over those with double center-embedded, object-extracted clauses, as in (16b). Furthermore, the network was able to make use of semantic constraints to facilitate word prediction in center-embedded sentences.

(16a) Tinman hears tiger that sees witch that tames lion.

(16b) Witch that tiger that tinman hears sees tames lion.

While this is an interesting finding, it is important to note that this does not necessarily imply a general preference for right-branching over center-embedded sentences for simple-recurrent networks. The sentences used in training and testing the network, as well as those in similar empirical studies, confound the location of the relative clauses with the extraction type of the relative clauses. The center-embedded sentences also happened to be object-extracted while the right-branching ones were subject-extracted. Object-extracted relative clauses are particularly hard because of the non-canonical word orderings they create, such as the three consecutive verbs in (16b). Thus, the model may be demonstrating a preference for subject-extraction, rather than for right-branching.

Finally, we might question whether a comprehension network would show the same preferences as a prediction network. The predictor has the advantage that it can forget information once that information becomes irrelevant. This is the principal explanation for why such a model prefers right-branching subject-relatives, since most of the information about the sentence can be discarded as it proceeds. On the other hand, a comprehender must remember all important semantic information at least until the end of the sentence. This may tend to weaken or possibly reverse any preference for right-branching sentences. These and other issues pertaining to relative clauses are discussed further in Section 3.2 and Chapter 11.

Chater and Conkey (1992) compared Elman's SRN training procedure to a more complicated variant, backpropagation through time (Rumelhart et al., 1986), which extends the propagation of error derivatives back to the beginning of the sentence. Not surprisingly, they found that backpropagation through time, which is slower and considerably less "biologically plausible" produces better results. Backpropagation through time is used in the CSCP model to similarly aid learning.

Christiansen (1994) tested the ability of SRNs to learn simple languages exhibiting three types of recursion: counting recursion[3], center-embeddings, and cross-dependencies, which exceed the power of a context-free grammar. However, any results are questionable since these experiments resulted in rather poor learning, with networks not even performing as well as statistical bigram models and sometimes worse than unigrams. It would be worth re-examining the methods used to train those networks. In a second experiment, Christiansen extended the language used by Elman (1991) to include prepositional phrases, left recursive genitives, conjunction of noun phrases, and sentential complements. One version of the grammar could produce center-embedded sentences and a second version cross-dependencies. In general, the networks performed rather well on these languages and exhibited behaviors that largely reflect human comprehension performance on similar sentences. **Christiansen and Chater (1999b)** extended these results and provided more detailed comparisons with human performance.

Finally, **Tabor, Juliano, and Tanenhaus (1997)** performed a number of experiments comparing human and network reading times on sentences involving structural ambiguities. Although the network used in these studies was just a simple-recurrent prediction network, reading times were elicited using a novel "dynamical system" analysis. Essentially, the hidden representations that appear in the network at various stages in processing sentences are plotted

[3]Counting recursion involves sentences composed of a sequence of symbols of one type followed by an equivalent number of symbols of a second type without any further agreement constraints.

in a high-dimensional space. These points are treated as masses that exhibit a gravitational force. To determine the reading time of the network on a particular word, the network's hidden representation for that word is plotted in the high-dimensional space and then allowed to gravitate among the attractors until a stable state is reached. The settling time is taken as a proxy for reading time. Although this test-mass process settling was intended to be a proxy for a true dynamical system that actually settles into a stable state, no experiments were performed to demonstrate that this is a reasonable simplification of such a model.

2.4 Production

Sentence production has received far less attention than parsing or comprehension in the symbolist community. This may be largely due to the emphasis on parsing in that tradition. If viewed simply as the inverse of parsing, or deriving a sequence of words from a higher-order representation of sentence structure, production is a simple process and can potentially be accomplished in a symbolic framework through the application of a few deterministic rules. However, true production is a mapping from an intended meaning to a sequence of words or sounds, which is a very hard problem. Production involves such diverse problems as choosing words to convey the appropriate message, selecting the correct morphemes to obey syntactic and agreement constraints, and modeling the listener's knowledge to allow the speaker to avoid redundancy, provide an appropriate level of information, and produce syntactic forms and prosodic cues that emphasize important parts of the utterance and avoid ambiguity.

Producing the appropriate phrasing depends on sensitivity to nuances of meaning that are difficult to capture in a symbolic system (Ward, 1991). Thus, some researchers have begun turning to connectionist approaches to modeling production. However, most connectionist language production models have so far been restricted to the word level, dealing with lexical access and phoneme production, rather than sentence-level phenomena (Dell, 1986; O'Seaghdha, Dell, Peterson, & Juliano, 1992; Harley, 1993; Dell, Juliano, & Govindjee, 1993). This section considers the most notable sentence production networks.

Kalita and Shastri (1987, 1994) focused on the problem of producing the words in a sentence given the thematic role fillers and indications of the desired voice and tense. Their model, which is a rather complex localist network, is able to produce simple SVO sentences in active or passive voice and in several tenses. In order to ensure that constituents are produced in the proper order, the model uses *sequencer* units to inhibit nodes once they have performed their duty. A special mechanism is included to allow the noun-phrase production component to be reused. Because of the complexity of hand-designing a localist network of this type and of representing thematic roles in multi-clause sentences, it is unlikely that this model could easily be extended to more complex sentences, particularly those with recursively nested structures. The model does not seem to exhibit any properties that transcend those of symbolic systems.

Gasser (1988) (see also Gasser & Dyer, 1988) described a significantly more ambitious localist model that produces sentences using elaborate event schemas. The model, known as the Connectionist Lexical Memory, is based on interactive-activation principles. Bindings to syntactic roles are encoded with synchronized firing, as in temporal synchrony variable binding (Henderson, 1994a). Sequencing is accomplished using start and end nodes for each phrase structure, which are somewhat similar to the sequencer units in Kalita and Shastri's model. Gasser's model is designed to account for a wide range of phenomena, including priming effects, speech errors, robustness given incomplete input or linguistic knowledge, flexibility in sequencing, and transfer of knowledge to a second language. The model is also able to parse sentences using the same sequencing mechanism as for generation but may not be able to handle lexical ambiguities or garden paths. However, the model was only applied to simple clauses and noun phrases and does not produce recursive structures involving long-distance dependencies. Again, it is not clear whether such a localist model could be scaled up to handle more complex sentences.

The third major localist production model was by **Ward (1991)**. This was intended to be "more connectionist" than the previous attempts, relying on a truly interactive settling process and avoiding the need for binder units. Ward described previous models as essentially serial in their processing. His model, like Gasser's, was designed to handle both Japanese and English. One major limitation of the model, which may apply to the others as well, is that the network structures used to represent the intended meaning of the utterance are built on a sentence-by-sentence basis. Although the model is apparently able to produce a broader range of sentences than the previous attempts, it is still unable to handle agreement, anaphor, and relative clauses. Ward acknowledged that a primary drawback of the model is the difficulty of extending it in all but the most trivial ways, and he recognized the need for a learning mechanism.

The inability to learn or to handle complex structure appears to be inherent in localist production models, which should not be surprising since these systems tend to be rather transparent implementations of classical finite-state machines. However, while not truly context-free, natural language is certainly pseudo-context-free or even pseudo-context-sensitive in that it allows a limited amount of recursion. For a simple, localist finite-state machine to capture such recursion, it would require replicated structure, which would presumably be a serious hindrance to generalization. We therefore turn to models that make use of distributed representations with the hope of overcoming these problems.

Kukich (1987) was interested in the ability of a network to learn to produce stock market reports given the day's activity. He doubted the ability of a single network to learn the entire task and thus trained one network to associate units of meaning, or sememes, with morphemes and another network to re-order morphemes. Sememes were represented as a series of slot fillers encoding such information as the type of trading activity and the direction and duration of any change. The output of the first network was an unordered set of word stems and suffixes, which could be produced accurately 75% of the time. The morpheme-ordering network did not actually produce morphemes sequentially but used a slot-based encoding of order. The results of these simulations left considerable room for improvement but were encouraging given the early state of connectionism.

We have already discussed the comprehension and production models of **Miikkulainen (1990b)** and **Miikkulainen and Dyer (1991)**. These were trained to produce either sequences of sentences based on a slot-filler representation of a story or multi-clause sentences based on a slot-filler representation of its clauses. So far this work has been restricted to fairly simple domains. The nature of the representations used appears to limit the ability of the system to be scaled up to more natural languages.

Finally, **Dell, Chang, and Griffin (1999)** were specifically interested in the phenomenon of structural priming, which leads speakers to preferentially produce sentences of a particular form, such as passive rather than active voice, if they have recently heard or produced sentences of similar form. Dell et al. hypothesized that the mechanism that results in structural priming is the same procedure used to learn production. Their model takes a representation of the sentence's propositional content and produces the words in the sentence sequentially. While it was intended to be an SRN, the recurrent portion of the model was not actually implemented, but was approximated by a symbolic procedure. Propositional content was encoded using a slot-based representation consisting of localist representations of the agent, patient, recipient, location, and action. Therefore, the model was able to produce only simple sentences with a limited range of prepositional phrases.

Based on whether the agent or patient received greater emphasis, the model was trained to produce either active or passive sentences. It was also able to convey recipients using a prepositional phrase or a dative. The model learned to produce sentences with 94% of the words correct. Based on an average sentence length of 4.8 words, we might estimate that this translates to about 74% of sentences being produced correctly. The model was able to match human structural priming data quite well. The main limitations of this model were that it was applied only to simple sentences, did not produce sentences as accurately as one might hope, and did not learn distributed context representations. The model presented in the current paper has similarities to that used by Dell et al. but amends some of its limitations.

2.5 Other language processing models

A few additional connectionist investigations of language that do not fit clearly into one of the above categories are worth mentioning.

Hanson and Kegl (1987) trained an auto-encoder network, known as PARSNIP, to compress sentences drawn from the Brown corpus (Francis & Kucera, 1979). Words were replaced by one of 467 syntactic categories, each encoded using 9 bits. Only sentences with fewer than 15 words were selected, eliminating most relative clauses. The input and output representations for the network comprised 15 slots, holding the syntactic categories of all of the words in the sentence at once. PARSNIP was trained using backpropagation to map from the input to the identical output through a smaller layer of 45 units. When trained on 10,000 sentences, the network was able to reproduce about 85% of the word categories correctly. The network performed at about the same level on novel sentences, indicating robust generalization. PARSNIP was reportedly able to fill in missing sentence constituents and correct bad constituents, and did so in a way that did not follow first-order statistics. It could handle single embeddings, despite their not having been trained, but not double embeddings or some sentences that violate English word order constraints. Although Hanson and Kegl acknowledged that auto-association is not a reasonable model for language acquisition, the importance of this work, as of the prediction models, was its demonstration that distributed networks can learn to be sensitive to higher-order structure merely through exposure to surface forms and can generalize that knowledge in productive ways.

Allen (1987) performed a number of small studies of language using backpropagation networks. In one experiment, a network was presented sentences containing pronouns referring to nouns appearing earlier in the sentence and was trained to identify the location of the original noun. Although it is not clear how well the network could actually perform the task, it was able to make use of semantic properties and gender in resolving some references. A second experiment involved training a network to translate from English to Spanish surface forms. The sentences dealt with a single topic, were limited to 11 words in length, and were presented to the network statically. A multi-layer feed-forward network was able to translate the sentences on a novel transfer set with an average of just 1.3 incorrect words. Although these early experiments were relatively simple, they were indicative of the ability of networks to learn complex language-related tasks.

Finally, **Chalmers (1990)** demonstrated that connectionist networks, while able to construct compositional representations through mechanisms such as the RAAM (Pollack, 1988, 1990), can also operate directly on those representations in a holistic fashion without first decomposing them. Chalmers first trained a RAAM to encode simple active and passive sentences and then trained a second network to transform the structural encodings of an active sentence to that for the corresponding passive sentence. The transformation network was found to generalize quite well to novel sentences. This simple experiment demonstrated that networks can perform structure-sensitive operations in a manner that is not simply an implementation of symbolic processes. Furthermore, transformational operations performed on learned hidden representations can often result in better generalization than transformations performed on surface representations.

In summary, other than prediction networks which avoid the issue of meaning entirely, no connectionist sentence processing models have exhibited all of the main properties necessary to provide a plausible account of natural language acquisition. These include the ability to learn a grammar, to process a sentence sequentially, to represent complex, multi-clause sentences, and to be naturally extendable to languages outside of the domain originally addressed by the designer.

Chapter 3

Empirical Studies of Sentence Processing

In order to evaluate the connectionist sentence comprehension and production (CSCP) model, it is necessary to compare its pattern of behavior with that of humans under similar conditions. The goal of the model is to model behavior not just on a particular type of sentence, but across a broad range of domains. The purpose of this chapter is to review many of the relevant psycholinguistic studies and to clarify the most interesting and most reliable results. Findings tend to vary greatly across experiments. Phenomena for which the findings are inconsistent or for which the experiments were poorly controlled will not provide a firm basis for evaluating the model. Therefore, the emphasis here is less on the actual numerical results than on the qualitative patterns of data. In some cases, new experiments have been suggested that might help address open issues.

3.1 Introduction

By and large, experiments were not included in this review if they were not easily addressable given the current constraints on the CSCP model. The most useful experiments are those that pertain specifically to comprehension or production at the level of individual sentences. An exception to this is that a number of studies were included that focus on the effects of context in resolving ambiguities. These are considered separately in Section 3.7.

Another criterion for inclusion is that the experiment deal with a common or important structure in the language. There are two reasons for this. Philosophically, it seems best to address a general model, such as the present one, toward the big issues first, before tackling more minor or rare phenomena. More practically, common features of the language are easier to handle in an appropriate way given our current resources. Because connectionist models are potentially quite sensitive to the statistics of their environment, it is important that factors relevant to the experiments of interest are carefully controlled for in the model's training language. If the structures being studied are too rare, it will be difficult to obtain accurate statistics and the limited exposure of the model to those structures may adversely affect the results. Such phenomena will be difficult to address appropriately until we have larger and more varied natural language corpora so that the relevant environmental input to the learner can be understood.

Prosodic information, including phrasing, intonation, and stress, undoubtedly plays another important role in aiding comprehension. Although one would expect a connectionist architecture to naturally take advantage of prosody, this too will not be available to the current model and its effects will not considered here in sufficient length or detail.

However, all this is not to say that the model will only be applied to simple constructions or simple effects. On the contrary, the phenomena to be addressed by the model involve some of the most complex and interesting aspects of language, including nested relative clauses, passives, main verb/reduced relative ambiguities, sentential complements, prepositional phrase attachment, subordinate clauses, structural priming, and the influence of semantics and pragmatics on syntactic processing.

Although most psycholinguistic data is collected in the service of advancing one theory over another, the aim of this chapter is to summarize the data with minimal interpretation. At this point, only scant attention will be paid to how well the data fits particular models or theories.

3.1.1 Testing methods

Evaluating the performance of the model involves comparing its behavior with that of humans faced with a similar task. This necessarily brings up the issue of how performance is to be measured in both humans and in the model. During comprehension, we are generally concerned with the overall ability to comprehend the sentence and the points where this breaks down. Sentence comprehensibility has been tested in a number of different ways, and there may well be just as many interpretations of the term *comprehensibility*. This section briefly describes the most common methods of assessing comprehensibility, most of which will be referred to in later discussions.

To begin with, I adopt the principle that the comprehensibility of a sentence is the degree to which it conveys the intended information when heard or read in its natural context. As far as the network is concerned, comprehension is most directly evaluated by its answering simple questions about the component propositions of the sentence. Therefore, the use of similar questions is the preferred method for testing human subjects' offline performance, and this method has indeed been used in a few studies. However, most such studies tend to use true/false questions, while the model's performance can be best diagnosed using fill-in-the-blank or multiple-choice questions.

A similar, though perhaps more taxing, measure requires the subjects (Ss) to *paraphrase* sentences, which might involve translating center-embedded constructions to right-branching ones or to their constituent clauses. The problems of this method are that it is not a very natural activity for most Ss, the task tends to be underspecified, and it may be inconsistently performed. It is likely that the sequential production demands of paraphrasing place a significant memory burden on the subject beyond the demands of just comprehending and retaining the sentence.

A less direct measure of offline comprehensibility is *delayed repetition*, where it must be assumed that sentences are easier to remember and reproduce when they are comprehensible. In this case, sentences are generally equated for number of words used, which limits the range of phenomena that can be studied under full control.

The method used frequently by Gibson (Gibson & Thomas, 1995, 1997; Gibson, 1998) and others involves direct *comprehensibility ratings*. That is, Ss do not demonstrate their ability to comprehend the sentence but simply try to report how difficult it was to understand. This is convenient from a practical standpoint because large groups of Ss can be run simultaneously and analysis is easy, but it has a number of potential problems. Ss may not be using the complexity scale consistently, although rescaling of ratings on a subject-by-subject basis may be possible. More importantly, they may not be accurately reporting comprehensibility, as defined by the ability to understand the information conveyed by the sentence. It is likely that other factors enter into comprehensibility ratings including the apparent syntactic complexity of the sentence, the frequency of its syntactic structures, and its semantic plausibility.

There is a danger in equating apparent and real comprehensibility. It has frequently been argued that repeatedly right-branching sentences such as (17) do not cause comprehension difficulties and thus appear to be comprehensible. However, although there is no point at which the sentence gives the impression of ungrammaticality, actually comprehending the sentence, that is, understanding and remembering the information in it, is not easy. As Blaubergs and Braine (1974) found, after hearing a sentence of that form, Ss answered questions such as "Who chased the lawyer?" incorrectly nearly half of the time. So the main problem of elicited comprehensibility ratings is that the values reported by Ss may systematically differ from less subjective measures of comprehension.

(17) The horse bit the cow that chased the lawyer that frightened the rabbit that examined the evidence that stunned the jury.

In other studies, Ss have been asked to provide *grammaticality ratings*. These will differ from a comprehensibility ratings to the extent that Ss understand the difference between grammaticality, as traditionally defined in linguistics, and comprehensibility. In practice, the responses of naive Ss actually do look somewhat like comprehensibility ratings and suffer from the same problem of measuring apparent rather than actual comprehensibility.

A more indirect measure of comprehension difficulty, but one that has seen extensive use, is *reading time*. The primary advantage of reading time, and the reason it is so favored by many researchers, is that it is believed to provide a window into the online processes of comprehension. The time that a subject takes to read a word or phrase is presumed to reflect the difficulty of incorporating it into an iteratively constructed representation of sentence structure or meaning.

Unfortunately, reading time is not the best task on which to evaluate the CSCP model. The network is intended to be a model of comprehension and production of spoken language, in which the listener has little control over the rate of presentation. It is thought that our reading and writing abilities are built on top of the more basic verbal communication system. Ideally, we would use data from verbal tasks in assessing the model. However, due to the lack of good online measures of auditory comprehension, the majority of interesting data comes from reading. Section 6.5 discusses this problem further and presents a method for extracting reading times from the current model, as well as discussing a novel theory of sentence comprehensibility and its relation to reading time.

Reading time experiments are typically either self-paced or involve eye-tracking devices. Self-paced reading can proceed a single word at a time, two words at a time, or a phrase at a time. Displays can either be cumulative, in which words remain visible once they appear, or use a moving window over obscured text. In the latter case, as each new word or phrase appears, the previous one is hidden from view to prevent re-reading.

Although self-paced reading does often produce similar results, the consensus among most researchers seems to be that eye-tracking studies are preferable. Eye tracking provides a more natural reading experience, resulting in considerably faster reading times. It also allows subjects to use the peripheral information normally available to readers, and there is less worry that the choice of segmentation of the display into either words, word pairs, or phrases may affect the results. When using an eye-tracker, subjects are free to move their eyes as they wish and will frequently jump back to review earlier parts of the sentence. This makes analysis more difficult. Experimenters typically analyze first-pass reading times separately from overall reading times. Because the CSCP model is forced to process sentences in a single pass, the first-pass reading times may serve as a better basis than total reading times for evaluating the model.

Word length is well known to have an effect on reading time. Because researchers are mainly interested in reading time as a proxy for online comprehension difficulty, any effect of word length is of less interest. Therefore, it is customary to convert raw reading times into *residual* reading times. This is done by deriving a linear regression for each subject to predict reading time as a function of word length. The predicted time is then subtracted from the raw reading time for each word to produce a residual time.

In addition to reading time, some even more indirect measures of comprehension have been used, including response time in initial-phoneme monitoring and lexical decision. While these methods have been validated to the extent that they give similar results to direct measures on tasks involving major comprehension differences, it is not clear that they reliably reflect comprehension difficulty across a wide range of tasks.

3.2 Relative clauses

The one structure in language that seems to have received the most attention from psycholinguists is the relative clause (RC). Although relative clauses have been studied for several decades now, the main findings on them remain poorly understood, and there are still many open questions regarding human comprehension of them. Nevertheless, there seems to be sufficient data available in the literature to allow processing of relative clauses to be a main area of evaluation for the network model.

A principal goal of some of the early studies of the 1960's and 1970's was to test the Miller and Isard hypothesis (Miller & Isard, 1964), that self-embeddings,[1] in particular, ought to cause problems for a finite device due to their recursiveness. This led to a continuing interest in the complexity of relative clause constructions, as it is easy to embed one RC within another.

Another early proposal was the *derivational theory of complexity* (Fodor & Garrett, 1967), suggesting that sentence complexity should be related to the number of transformations in the mapping from deep to surface structure. As evidence mounted in opposition to these theories, more detailed metrics of sentence complexity were devised including Kimball's *principle of two sentences* (Kimball, 1973), Gibson's *thematic-role-based theory* (Gibson, 1991), *syntactic prediction locality theory* (Gibson, 1997), and *dependency locality theory* (Gibson, 2000), and Lewis's NL-Soar model which hypothesizes that only two or three syntactic relations of the same type can be represented in memory (Lewis, 1993).

One feature shared by nearly all symbolic models of sentence processing is an insensitivity to semantic and pragmatic constraints. Although most researchers do acknowledge that semantic information can facilitate human compre-

[1] A self-embedding is any structure contained within another structure of similar type. A common example is an object-extracted relative clause contained within another object-extracted relative clause.

Sentence Types	
C	center-embedding
R	right-branching
O	object-relative
S	subject-relative
OS	subject-rel. in an object-rel.
O^2	object-rel. in an object-rel.
O^n	n nested object-relatives
$'$	reduced
$''$	reduced and untensed
Presentation Methods	
V	visual (written)
A	auditory (spoken)
S	speeded word-by-word
T	self-paced word-by-word
Evaluation Methods	
Q	question answering
P	paraphrasing
R	repetition
C	complexity judgment
M	phoneme monitoring
L	lexical decision
G	grammaticality decision

Table 3.1: Codes used in describing sentences and experimental procedures.

hension, such graded constraints are not allowed for explicitly in their models. This is for a variety of reasons. Graded constraints cannot easily be incorporated into many symbolic architectures. Even if the architecture permits semantic constraints, they are hard to quantify and control and they complicate models, reducing the ease of formulating predictions.

As a possible result of this, many experiments on relative clauses have been designed to eliminate semantic information. Typically, sentences are used in which every noun is a plausible subject or object of every verb, increasing the chance of confusion. As a result, sentence structures that may be comprehensible in practice, where both context and semantic constraints within the sentence are strong, might appear incomprehensible under experimental conditions. Furthermore, there are few tests of the actual effect of semantics and pragmatics on the variety of sentence types. This is particularly lamentable because the CSCP model is expected to be quite sensitive to such factors.

This section will review the major empirical findings on relative clauses. The studies discussed here are summarized in Table 3.2, which should serve as a quick reference in comparing studies. Table 3.1 explains the codes used to classify the experiments. The individual studies are discussed and further summarized in Sections 3.2.1 through 3.2.5.

3.2.1 Single relative clauses

Sentences with a single relative clause have not received quite as much attention as multiply-embedded sentences because, by and large, people are quite good at comprehending them and thus all models that treat comprehension as a matter of success or failure would predict success. However, these relatively simple sentences will be an important point of evaluation for the CSCP model as they are common in everyday discourse. As comprehension models become more sophisticated, they should be able to explain the relatively small on- and offline differences in the comprehension of single-relative-clause sentences.

In addition to whether the relative clauses are reduced or are marked[2] by a relative pronoun, two main factors are at

[2]It is common practice in linguistics, or perhaps just in psycholinguistics, to use the term *marked* to mean ungrammatical or otherwise unusual. I will, however, be using *marked* to mean that which has a marking, in opposition to *reduced*. Some have referred to this as unambiguous versus ambiguous. But there are sometimes several ways in which ambiguities can be made unambiguous, not all of which involve marking, so I will avoid

Sentence Type	B66	S67	M68	FG67	HC70	H73	BB74	BK74	HEB76	LB77	HO81	F83	KJ91	GT97	G*ip
CS						SR		AQ	AMP		VQ	TL	T		TQ
CO			VG			SR	AQ	AQ	AMP	AP	VQ	TL	T		TQ
RS			VG			SR	AQ	AQ	AMP						TQ
RO						SR		AQ	AMP						TQ
CS''									AMP						
CO'									AMP						
RS''									AMP						
RO'									AMP						
CS^2			VG												
CSO															
COS														VC	
COS''														VC	
CO^2		VP	VG	AVP	AMP		AQ			AP				VC	
CO'^2				AVP	AMP										
CO^3	AVP		VG				AQ			AP					
CO^4			VG				AQ			AP					
CO^5			VG				AQ								
RS^2			VG				AQ								
RS^{3-5}			VG				AQ								
ROS														VC	
RO^2														VC	

Table 3.2: Sentence types used and experiments performed in some of the studies discussed here. Not all studies are shown in this chart. G*ip refers to Gibson et al. (in press).

work in sentences with a single relative clause: whether the clause is *center embedded* (modifying the matrix subject) or *right branching* (modifying the matrix object), and whether the clause is subject- or object-extracted (also called subject- or object-focused).[3] For convenience, subject-extracted RCs will be referred to here as subject-relatives and object-extracted RCs will be referred to as object-relatives. Sentences (18a) and (18b) both have center-embedded RCs while (18c) and (18d) have right-branching RCs. (18a) and (18c) have subject-relatives while (18b) and (18d) have object-relatives.

(18a) CS: The dog that bit the cat chased the bird.

(18b) CO: The dog that the cat bit chased the bird.

(18c) RS: The dog bit the cat that chased the bird.

(18d) RO: The dog bit the cat that the bird chased.

Several of the early studies that purported to show that center-embedded sentences are harder than right-branching sentences actually confounded this difference with the clause-type distinction, comparing CO sentences to RS sentences. This has led to some confusion about these sentence types and a lingering belief that center-embedded sentences, in general, are more difficult than right-branching sentences.

One such study was **Blaubergs and Braine (1974)**. Blaubergs and Braine compared singly and multiply-embedded CO^n and RS^n sentences, with the number of embeddings, n, ranging from 1 to 5. The sentences were semantically neutral (each noun could reasonably be the subject or object of each verb), and matched pairs of CO^n and RS^n were formed. Sentences were presented auditorily and Ss answered a single fill-in-the-blank question following each sentence. Prior to testing, Ss were trained on the question answering task with sentences of increasing order of complexity. Although there was a main advantage for RS^n sentences over CO^n sentences, which will be discussed in the next section, Ss performed nearly perfectly on CO and RS sentences and there was even a slight but non-significant advantage

using such terms to distinguish marked from reduced conditions.

[3]Center-embedded RCs are more appropriately termed subject-modifying while, right-branching ones are more appropriately termed object-modifying. However, distinguishing between subject-modifying and subject-extracted or object-modifying and object-extracted can get very confusing, so those terms are avoided here.

for CO sentences.

In **Marks (1968)**, Ss were asked to rate the grammaticality of sentences composed of from 1 to 5 center-embedded or right-branching clauses. It is presumed that center-embedded clauses were all object-relative and right-branching clauses were all subject-relative and that sentences were presented visually. Unfortunately, this study sheds little light on singly embedded sentences, as they were almost always judged to be perfectly grammatical. It would appear that grammaticality rating and question answering following training are not sufficiently sensitive to detect differences in single RC sentences. However, in these two experiments, single RC sentences were presented along with a majority of much more difficult items. Under such conditions it is likely that subjects will raise their standards for ungrammaticality and read sentences more slowly and carefully, resulting in better than normal comprehension performance. As a result, differences between single RC sentences may not appear, as they may under other conditions.

Holmes (1973) tested Ss on sentence recall following rapid word-by-word visual presentation. Because earlier studies had shown effects of semantic plausibility on recall, it was believed that comprehension plays a role in this task and that recall success may reflect comprehensibility. Several sentence types were studied, but the ones of interest to us are the center-embedded sentences, half CO and half CS, and right-branching sentences, half RO and half RS. Although they were not equated for meaning, all sentences had 10 words, and attempts were made to equate sentence types for overall naturalness and plausibility. Holmes found that CO/CS sentences (mean 8.46 words recalled) were significantly easier than RO/RS sentences (7.42 words recalled).

Although the two types of sentences were independently judged to be very similar in semantic plausibility, center-embedded sentences were judged to be slightly more natural, 3.25 versus 3.14 on a 0-4 scale. This could be considered either a confounding or an explanatory factor. Although this study appears to provide strong evidence that, averaged over embedding type, single center-embedded sentences are easier than single right-branching sentences, recall after rapid serial visual presentation must be a regarded as a rather indirect measure of comprehensibility.

Baird and Koslick (1974) used auditory presentation and fill-in-the-blank questions to study comprehension of CS, RS, CO, and RO sentences. All nouns referred to humans and semantic constraints were reduced by using only noun/verb pairs known to be non-associates. Sentences were presented twice before testing. Averaged across question types, error rates for the four sentence types were CS:17.5%, RS:23.8%, CO:43.8%, RO:40.0%. It is interesting that error rates were so high across the board. Overall, subject relatives were much easier than object-relatives. However, center-embeddings were not significantly easier than right-branchings, and there was a hint of an interaction between location and clause types, with the CS being the easiest and the CO being the hardest. Because there were relatively few items in this experiment, the statistical power was not strong. The results for this experiment are summarized in Figure 3.1.

Hudgins and Cullinan (1978) studied single relative clauses using a sentence elicited imitation task. Ss listened to a recorded sentence and then tried to repeat it verbatim. Based on error rate, this study found a significant advantage for subject-relatives over object-relatives, both for short sentences and for longer sentences involving adverbials and other filler material. There was also a consistent advantage for center-embedded over right-branching clauses, but this was not a significant difference in all cases. The authors also tested reduced relatives which had only a small effect that was inconsistent between response latency and error rate.

A more complete study of single relative clauses is that of **Hakes, Evans, and Brannon (1976)**, in which Ss listened to sentences and performed phoneme monitoring. This involves responding to a particular phoneme if it occurs at the start of a word. The target words, when they appeared, were always located in the embedding. After hearing the sentence, subjects were asked to paraphrase it. In the first experiment, test sentences were either CO or RO. These occurred in matched pairs, in which the component clauses were the same but were rearranged to produce different structures and meanings. The sentences were all complex in ways other than the relative clause, including other main and subordinate clause types. Relative clauses were also either reduced or marked, but the reduced sentences will be considered in Section 3.2.3.

The results indicate a marginally significant advantage for center-embedding. Paraphrasing accuracy, as measured by percentage of clauses accurately represented, was 45.1% for CO and 40.9% for RO. This was also reflected in the phoneme monitoring speed which was 2.00 for CO versus 1.93 for RO, where a higher number indicates a faster response. The second experiment was similar to the first except that subject-relatives were used and embedded verbs were in the past progressive tense (*[who were] running*) to permit reduction. Again the unreduced case revealed a significant advantage for center-embedding with a paraphrasing accuracy of 67.9% for CS versus 62.5% for RS.

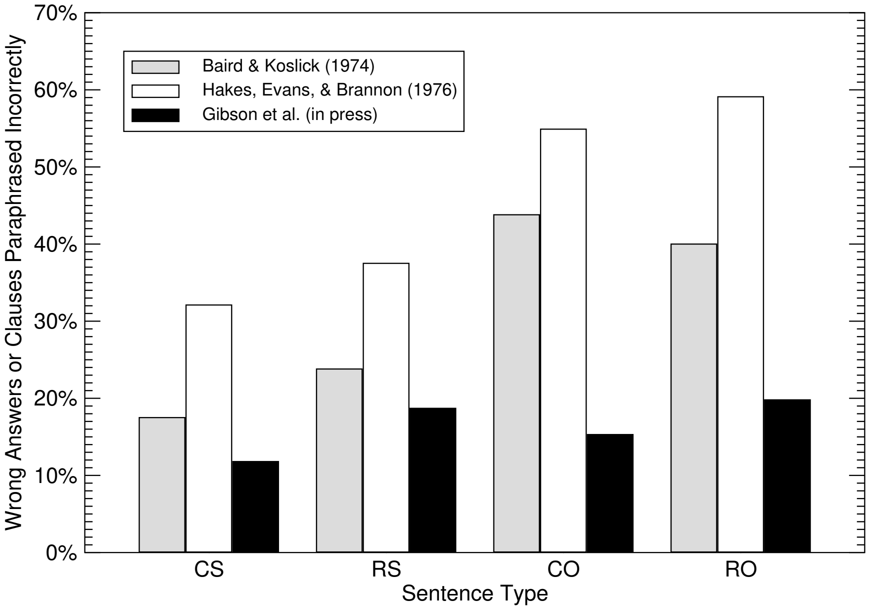


Figure 3.1: Relative clause paraphrasing and comprehension results from three studies.

The phoneme monitoring results, however, were not significant. Although the authors did not perform a statistical test because the experiments were conducted separately, it is clear from the large change in paraphrasing scores that subject-relatives were significantly easier than object-relatives. The results for this experiment are also summarized in Figure 3.1.

Holmes and O'Regan (1981) used an eye-tracker to study the reading of French sentences. After each sentence was read, it was removed and subjects answered a yes/no question. Sentences were in either CS or CO form. The error rate for questions concerning the main clause was 8.3% for CS and 12.5% for CO. Error rates for questions about the relative clause were higher, being 13.2% and 22.9%, respectively. This indicates that subject-relatives are easier than object-relatives in French as well as English.

Ford (1983) used word-by-word lexical decision as a measure of reading difficulty. Once again, CO and CS sentences were compared. These were arranged in pairs matched for the words used in the relative clauses. As a result, the relative clauses in the two conditions were roughly opposite in meaning, as in Hakes et al. (1976). Lexical decision reaction times were significantly longer for object-relative sentences only on the relative clause verb, the main clause verb, and the main object determiner. This once again confirms that single object-relatives are more difficult than single subject-relatives.

King and Just (1991) performed a self-paced reading study in which Ss had to remember the last word in each of a series of three sentences. The CO and CS sentences under investigation appeared as the second or third in the series. Reading times were significantly longer during the relative clause in the object-relative sentences.

Finally, **Gibson, Desmet, Watson, Grodner, and Ko (in press)** conducted two self-paced reading experiments on singly-embedded RCs. In the first, CS, RS, CO, and RO sentences were tested using self-paced reading and true/false question answering. In one set of conditions, the sentences were in the usual form, while in another set of conditions the sentences were embedded within a sentential complement. The latter, embedded, condition will be ignored here for now. Gibson et al. found two main effects: center-embedded RCs were faster to read and easier to comprehend than right-branching ones and subject-relatives were easier to read and comprehend than object-relatives, although these effects were significant only in reading times. The comprehension results are shown in Figure 3.1.

A second experiment investigated the difference between restrictive and non-restrictive RCs. Restrictive RCs serve to identify the specific reference of the modified noun phrase, while non-restrictive RCs simply provide additional information about the noun phrase. Interestingly, they found that CO clauses were read significantly faster than RO clauses only when they were restrictive. However, this finding was not supported by complementary question-answering data. Subjects were a bit better, though not significantly so, in answering questions about CO sentences, with little effect of clause type.

Summary

Based on the evidence considered here, it seems clear that center-embedded object-relatives are more difficult than subject-relatives, as consistently shown in Baird and Koslick (1974), Hakes et al. (1976), Holmes and O'Regan (1981), Ford (1983), King and Just (1991), and Gibson et al. (in press), although not always significantly so. Likewise, Baird and Koslick (1974), Hakes et al. (1976), and Gibson et al. (in press) found that right-branching object-relatives are more difficult than right-branching subject-relatives.

The more controversial comparison is that between center-embedded and right-branching structures. Holmes (1973) found that, when subject- and object-relatives were mixed, center-embedded sentences were easier than right-branching ones. Baird and Koslick (1974) looked at the four types of sentences in isolation and found no significant effect of relative clause position, which may largely be attributable to small sample sizes. Hakes et al. (1976), on the other hand, did seem to find that CS sentences were easier than RS, and CO were easier than RO, although that exact statistical test was not performed. Gibson et al. (in press) found an overall advantage for center-embedded over right-branching structures in both reading times and question answering. While the clause position effect may not be as strong as the clause type effect, it seems to be a reliable result.

One might wonder, however, whether there is a detectable difference in the ease of processing center object-relatives and right subject-relatives. The grammaticality decision test of Marks (1968) and the practiced question answering of Blaubergs and Braine (1974) were not able to detect a difference. Gibson et al. (in press) found non-significant preferences for CO over RS in both reading times and question answering. However, Baird and Koslick (1974) and Hakes et al. (1976) found what are certainly significant advantages for RS over CO. It seems that the CO/RS comparison involves an object-relative disadvantage counteracted by a center-embedding advantage. Which effect is stronger may depend on the specific items or methods used.

3.2.2 Nested relative clauses

Although some interesting effects may be observed with singly-embedded sentences, psycholinguists have historically paid more attention to multiply embedded sentences which lie at the fringe of comprehensibility. However, as we will see, studies of multiple embeddings often confound the center/right difference with the subject/object-relative difference. In a way, this is unavoidable as it is not possible to construct truly nested relative clauses unless all but the bottom-most are object-relative, unless ditransitives or prepositional phrases are used. However, we must be careful what generalizations are drawn from such studies. Unfortunately, the space of possible multi-clause sentences is quite broad and thus difficult to control, and our understanding of the comprehensibility of deeply-nested sentences remains woefully incomplete.

One of the earliest and best-known studies is that of **Miller and Isard (1964)**, who asked Ss to try to memorize and repeat verbally presented sentences. All sentences were 22 words long and contained four relative clauses, which comprised some mixture of center object-relatives and right subject-relatives. Ss heard and attempted to repeat each sentence five times and were evaluated on the number of words recalled in the proper order. Although statistical tests were not reported, it appears that performance generally declined with more center-embeddings. However, this did not occur in a smooth fashion. Sentences with 0 and 1 embedding were indistinguishable, as were those with 3 and 4 embeddings, with 2-embedding sentences falling somewhere in the middle. Nevertheless, on the basis of this experiment we cannot conclude much about the center versus right distinction as it is likely that these results are driven by the subject/object-relative difference.

Blumenthal (1966) questioned whether subjects even view multiply-embedded sentences as grammatical or whether they view them as distortions of a simpler and/or more common sentence type. In his experiment, Ss were shown CO^3 sentences and asked to reformulate them as semantically equivalent RS^3s. They were given unlimited exposure to the sentences but apparently no practice or feedback. All nouns referred to people and semantic constraints were rather weak. In only 26% of the sentences did Ss appear to understand the embedded structure and in only 15% of the cases were the verbs and nouns matched correctly. In the other cases, Ss reportedly either perceived a single clause with a compound noun and verb or a series of clauses all modifying the matrix subject. It is clear that $\mathtt{CO}^3$ sentences are fairly unnatural and might even be thought ungrammatical, but this experiment seems to have been unfairly weighted against the Ss. Had subjects been given better instruction or feedback and had the sentences included some semantic constraints, the results may have been different.

In **Marks (1968)**, mentioned in the previous section, Ss were asked to judge the grammaticality of sentences of the form $\mathtt{CO}^{1-5}$ or $\mathtt{RS}^{1-5}$. Although single relative clauses were judged to be perfectly grammatical, more interesting effects can be seen with deeper embeddings. Extended right-branching sentences were considered somewhat ungrammatical, but in a way that did not correlate with the number of clauses. Extended center-embeddings were judged to be increasingly less grammatical, although one S was thrown out because he (correctly?) judged all of the self-embedded sentences to be perfectly grammatical. It does not seem that this experiment has much bearing on the issue of comprehensibility as it is simply not clear what Ss are doing when asked to judge grammaticality, especially when all sentences were grammatical. The fact that Ss rated right-branching sentences to be ungrammatical, and inconsistently so, indicates that they were probably not clear on the task.

Schlesinger (1968) compared reading rates of Hebrew sentences containing either nested or sequential parenthetical clauses. An example sentence (in English) with degree of nesting 3 is:

(19) The defendant's solicitor demanded, since he knew that the court would not, in view of the attempts revealed subsequently under cross-examination to mislead the police officers in the first stages of the inquiry, accept the defendant's statement, that the fact that his client was the head of a large family should be taken into account in giving the verdict.

Sentences with lower degrees of nesting used the same phrases but rearranged them. To encourage comprehension, some sentences included contradictory information and Ss were asked to identify the ones that did not make sense. Interestingly, reading rate was the same for sentences with degree of nesting 0, 1, or 2 and only a bit worse at degree 3. Accuracy in the contradiction monitoring task was not affected by nesting depth.

In a second experiment reported in Schlesinger (1968), Ss read Hebrew sentences, then decided whether each one was grammatical, and then answered 12 true/false questions about it. Sentences were composed of either 4 or 5 clauses, which were arranged to produce a maximum depth of embedding between 0 and 3 for the shorter sentences or 0 and 4 for the longer ones. Reading was clause-by-clause and non-cumulative, to prevent looking back at previous parts. Ungrammatical sentences were included as foils. All legal sentences were judged, on average, to be grammatical. The degree of perceived ungrammaticality increased significantly, although not consistently, with depth. Although more explicit guidelines were given in this experiment than in Marks (1968) regarding grammaticality decisions, some subjects did, contrary to instructions, report basing their judgments on semantic considerations in addition to syntactic ones.

Response errors to the true/false questions were quite low, and were under 21% for all sentence types. Interestingly, errors did not increase significantly with depth. The results of these two experiments indicate that embedding does not necessarily make sentences more difficult and "is a much less powerful variable than is commonly supposed." (p. 119) However, it should be acknowledged that these sentences were in Hebrew and contained a variety of clause types in addition to relative clauses. Thus, these experiments are not directly comparable to others reviewed here.

We also return to the previously mentioned study of **Blaubergs and Braine (1974)**, in which Ss heard semantically neutral $\mathtt{CO}^n$ and $\mathtt{RS}^n$ sentences and answered a single fill-in-the-blank question following each sentence. Prior to testing, Ss were trained on the question answering task with sentences of increasing complexity. During training, they were asked an exhaustive list of questions and did not move on to the next-harder level of sentences until they made no more than one mistake in two consecutive sentences. On test, performance decreased for both sentence types with increased level of complexity. There was an overall advantage for $\mathtt{RS}^n$ over $\mathtt{CO}^n$, but it was only significant with 3, 4, or 5 relative clauses. Interestingly, the decline for right-branching sentences was gradual, but there was a large gap in performance between $\mathtt{CO}^2$ and $\mathtt{CO}^3$ and then no further decline with more center-embeddings.

Larkin and Burns (1977) had Ss listen to $\mathtt{CO}^{1-4}$ sentences and report the constituent clauses. During training, they were given one example sentence at each level. Sentences were either semantically neutral or semantically supported, although semantic support appears to have been weak. The results, in terms of the number of pairs correctly reported are: $\mathtt{CO}$:1.69, $\mathtt{CO}^2$:1.25, $\mathtt{CO}^3$:1.09, $\mathtt{CO}^4$:1.04. Surprisingly, Ss were far from perfect on sentences with just one or two embeddings, producing about 15% errors. Larkin and Burns suggested that, in normal discourse, people must rely on such additional cues as context and stress patterns. Semantically supported sentences were a bit easier than

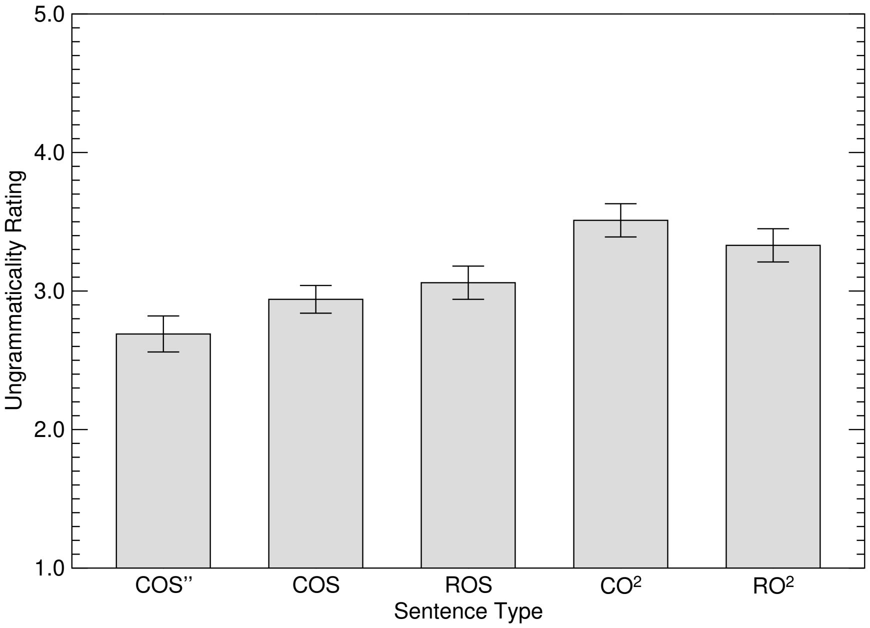


Figure 3.2: Results of the Gibson & Thomas (1997) complexity rating study.

unsupported ones, but only significantly so for $\mathtt{CO}^2$. Similar experiments were also performed using nested sequences of nouns paired with verbs, letters paired with digits, and digits paired with digits. Results for nouns paired with verbs were not significantly different from those for sentences, although there were no semantic effects for the non-sentence case. The letters and digits conditions proved easier than the others.

Gibson and Thomas (1995) report a study in which Ss read sentences and judged, on a 1 to 5 scale, how hard each sentence was to understand on the first reading. Although much of the experiment dealt with sentential subjects and complements, I will focus here only on the sentences with relative clauses, which took the forms $\mathtt{CO}^2$, COS, and $\mathtt{RO}^2$. Sentences were arranged in matched sets. Corresponding $\mathtt{CO}^2$ and $\mathtt{RO}^2$ sentences had approximately the same meaning, but to accomplish this the matrix clause of the $\mathtt{RO}^2$ sentences was put in the passive voice, a possibly confounding factor. The meaning of the COS differed from that of the $\mathtt{CO}^2$ in that the inner-most clause was reversed. In order to make the three verbs in each sentence more distinguishable, at least one was given an auxiliary verb, such as "had", "might", or "will." The second verb was also modified by an adverbial phrase. Unacceptability ratings for the $\mathtt{CO}^2$ and $\mathtt{RO}^2$ sentences did not differ significantly, although the mean for $\mathtt{RO}^2$ was lower. The preference for COS over $\mathtt{CO}^2$ approached significance.

Gibson and Thomas (1997) attempted to remove the confound of voice in the previous experiment by counter-balancing equal numbers of active and passive sentences and also added the missing ROS condition and a $\mathtt{COS}''$ condition in which the inner-most relative clause was reduced and untensed (a gerund), such as "neglecting" as opposed to "who was neglecting". The results of this experiment are shown in Figure 3.2. There was again no significant effect of extraction position, but there was a significant advantage for OS sentences over $\mathtt{O}^2$. Interestingly, there was also a significant advantage for the gerund, $\mathtt{COS}''$, over the non-reduced COS, a subject we will return to in the next section.

Summary

The experiments in this section can be roughly divided into four groups. Blumenthal (1966) and Larkin and Burns (1977) studied $\mathtt{CO}^n$ sentences. Blumenthal found that, under less than helpful conditions, naive Ss had considerable difficulty reading and rephrasing $\mathtt{CO}^3$ sentences, indicating that they were lacking in the grammatical knowledge necessary to parse them, rather than suffering from performance problems. This suggests that, despite claims by some linguists of rampant productivity, multiply embedded object-relatives may be ungrammatical in English.

Larkin and Burns studied paraphrasing in $\mathtt{CO}^{1-4}$ sentences and found that comprehension declined significantly with the number of embeddings. Given that single object-relatives are not easy and the fact that these studies involved rather unnatural conditions and the difficult task of paraphrasing, it is not surprising that Ss performed poorly.

In the second set of studies, Miller and Isard (1964) and Schlesinger (1968) examined sets of sentences with a fixed total number of clauses where the dependent variable was how many of the clauses were self-embedded rather than right-branching. The memorization and recall experiments of Miller and Isard suggest that a single center-embedding is no harder than an entirely right-branching sentence. Two center-embeddings are more difficult and three and four are more difficult still, but are similar to one another. However, the memorization task may not be a truly accurate measure of comprehensibility.

Schlesinger looked at reading rate while Ss attempted to detect contradictory material and found that depth of embedding had no effect on the comprehension task and only resulted in slower reading once three levels of embedding were reached. The second experiment used sentences with up to four embeddings and queried comprehension directly. Although Ss rated more deeply nested sentences to be less grammatical, depth of embedding had no effect on the ability to answer comprehension questions.

One reason that Schlesinger did not find strong effects of embedding may be that the sentences used in his study were quite semantically constrained, while those in most other studies were not. The fact that Schlesinger's sentences were in Hebrew and that they contained a variety of clause types reduces our ability to compare this directly to other work. If a conclusion must be drawn, it seems to be that, given a fixed set of clauses, sentences with deeper embeddings are likely to be somewhat harder. However, the effects of embedding depth are probably less important than other factors, such as whether the sentence is phrased in an awkward way or is semantically ambiguous.

Marks (1968) and Blaubergs and Braine (1974) directly compared CO^n and RS^n sentences. Marks asked Ss to judge grammaticality and found inconsistent effects for right-branching sentences but, for CO^n, judgments became increasingly worse with depth. However, there are several reasons to believe that grammaticality decisions are not an accurate reflection of comprehensibility, or even of grammaticality for that matter. Blaubergs and Braine (1974) directly tested comprehension by asking fill-in-the-blank questions and found that comprehension of CO^{3-5} sentences was significantly worse than of RS^{3-5} but that CO^2 was not significantly worse than RS^2. Although it is clear that deep center-embedded object-relatives are difficult, it is not clear how much of this is due to center-embedding and how much is due to awkward object-relative clauses.

Finally, Gibson and Thomas (1995, 1997) asked subjects to make grammaticality decisions on sentences with two relative clauses that modified either the matrix subject or the matrix object and which consisted of either two object-relatives or a subject-relative within an object-relative. As we might expect, based on the single clause data discussed in Section 3.2.1, there was no effect of position but the double-object relatives were judged less grammatical.

Unless some major experiments have been overlooked, it is clear that we have barely scratched the surface of issues involved in comprehension of sentences with multiple relative clauses. The results reported here are all consistent with the simple rule that *object-relatives are difficult*, with the corollary that having more of them makes things more difficult. Better controlled experiments will be necessary to determine if that is indeed the only factor at work here. Some suggestions are discussed in Section 3.2.6.

3.2.3 Reduction effects

A priori, it is not clear whether we should expect reduced relative clauses to be more or less difficult than marked (unreduced) RCs. On the one hand, reduced clauses are shorter and thus might place a smaller burden on the comprehension system. On the other hand, reduction could cause extended ambiguity. In addition to the well known main verb/reduced relative and sentential complement ambiguities, reduction can make nested object-relatives appear to be a compound noun, particularly if the sentence is spoken, as in (20a) and (20b).

(20a) The dog the cat the pig chased bit ran away.

(20b) The dog the cat the pig and the monkey ran away.

Fodor and Garrett (1967) tested paraphrasing following auditory presentation of CO^2 and CO'^2 sentences, where the reduced CO'^2 sentences lacked relative pronouns. Sentences were read with flat intonation, stress, and phrasing. Both the quality of the paraphrase and the time to begin producing it were recorded and the ratio of the two used as a measure of comprehension ease. Performance on the marked sentences was significantly better. When sentences were presented with normal prosody, scores improved in both conditions but, reduced sentences with prosodic cues were still harder than marked sentences without them. In a third experiment, reduced sentences contained pauses where the relatives would have occurred to equate presentation rate, but scores did not improve. Finally, the same results were obtained from written presentation.

Hakes and Cairns (1970) performed a paraphrasing and phoneme monitoring experiment on the same sentences and found, contrary to an earlier study (Foss & Lynch, 1969), that phoneme monitoring also reveals an advantage for marked sentences. It seems quite clear from these studies that, for center-embedded object-relatives, removal of the relative pronouns makes the sentences more confusing. Interestingly, this does not seem to be a result of marked object-relatives being more frequent in the language. As shown in Section 4.3, reduced object-relatives are more common than marked object-relatives by about a 2 to 1 ratio and the ratio is even greater for object-relatives modifying the subject and for nested RCs.

Two other studies that we have previously mentioned included reduced relative conditions. **Hakes, Evans, and Brannon (1976)**, already discussed in Section 3.2.1, tested phoneme monitoring and paraphrasing of both marked and reduced CS, CO, RS, and RO sentences. In order to permit reduction, subject-relatives were in the past progressive tense, *[who were] running*. The results are a bit complex. For subject-relatives, neither monitoring speed nor paraphrasing showed consistent or significant effects of reduction. For object-relatives, reduction hindered monitoring speed but improved paraphrasing. The non-effect of reduction for subject-relatives is understandable in that reduced past progressive relative clauses are reasonably common and unambiguous. It is also understandable that monitoring speed was slower for reduced object-relatives given the known difficulty of CO'^2 sentences. The paraphrasing results are troubling and a replication would be in order, but one could conjecture that the demands of the task forced the Ss to attend more strongly to the reduced object-relative sentences, which thus improved their paraphrasing.

Finally, as mentioned in the previous section and depicted in Figure 3.2, **Gibson and Thomas (1997)** found that sentences with a reduced past progressive tense subject-relative inside of an object-relative, COS'', were rated by Ss as less comprehensible than marked COS sentences. However, as mentioned earlier, comprehensibility ratings may include considerations of perceived syntactic complexity and thus not accurately reflect true comprehensibility. It is possible that subjects perceived that the reduced sentences had been "transformed" and thus decided that they ought to be more difficult to comprehend. The effect of reduction is an interesting issue that should be examined in more depth using direct measures of comprehension difficulty.

3.2.4 Semantic and pragmatic effects

One critical factor that has not received sufficient attention is the role of semantic constraints in facilitating comprehension of relative clauses. Because the early comprehension theories did not permit semantic effects, experiments designed to test these theories often attempted to eliminate any semantic biases. Other experiments used more natural sentences but either did not control for semantic effects or, at least, did not treat semantics as a factor. We have already mentioned the fact that the experiments of Schlesinger (1968) found no effects of embedding depth on comprehension while those of Miller and Isard (1964) did find depth effects in a sentence recall task. One possible reason for this is that Schlesinger's sentences, such as example (19) above, appeared to be actual natural sentences with fairly strong semantic constraints while those of Miller and Isard (1964) had only weak semantic constraints, as in (21).

(21) The book that the man that the girl that Jack kissed met wrote told the story that was about a nuclear war.

Stolz (1967) looked at whether Ss, given some training, could learn to comprehend multiply-embedded sentences. Ss read each sentence while they heard the experimenter recite it three times and were then to write the component clauses as simple sentences. The experiment began with 10 non-embedded practice sentences, on which Ss were given feedback. The next five practice sentences were in CO^2 form and subject groups differed in whether the sentences contained semantic constraints and whether feedback was provided. The last five test sentences were in CO^2 form without semantic constraints or feedback. The data, shown in Figure 3.3, reveals that subjects learned well only when feedback was provided and the training sentences were semantically neutral. This indicates that subjects can learn to process sentences with up to two embeddings, given sufficient exposure, but that they do not learn as well when they are not forced to attend to structure because of a lack of semantic clues.

The one study that actually attempted to isolate the effects of semantic constraints on comprehension was **Larkin and Burns (1977)**. In this experiment, discussed in Section 3.2.2, Ss paraphrased CO^{1-4} sentences. Two conditions

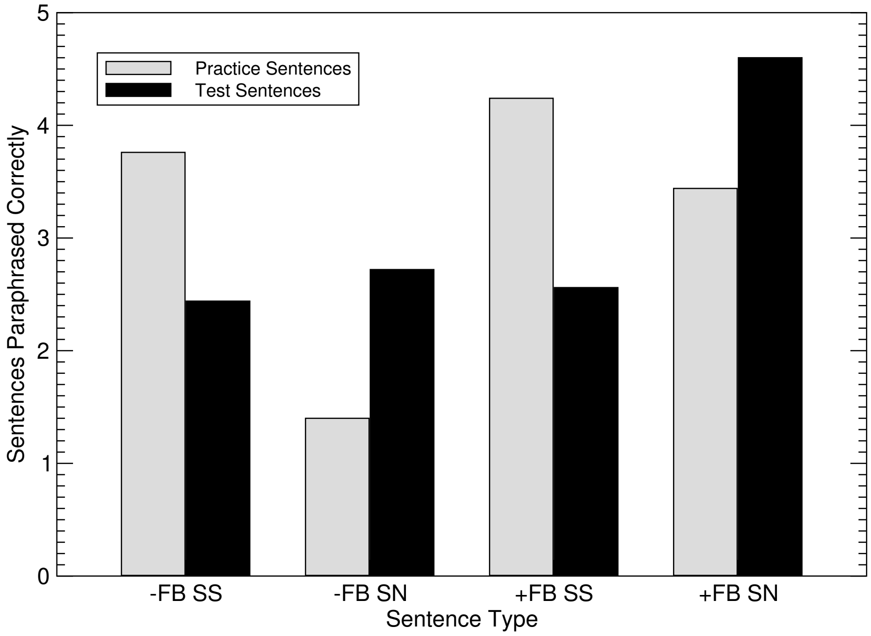


Figure 3.3: Paraphrasing results on CO^2 sentences from Stolz (1967). ±FB indicates whether feedback was provided during practice. Practice sentences were either semantically supported (SS) or semantically neutral (SN). Test sentences were all SN.

were explored. In the semantically neutral condition, each noun was a reasonable subject or object for each verb. Semantically constrained sentences permitted fewer misinterpretations but semantic relations alone were not sufficient to decode the syntax. (22) is an example of one of the most semantically constrained sentences.

(22) The wall that the tapestry that the curator viewed covered crumbled.

Although performance on the SS sentences was better on all sentences types, it was only significantly better on the most complex, CO^4, sentences. However, Larkin and Burns admit that semantic constraints were kept rather weak to allow comparisons with other conditions.

It is clear that there is a lot of work to be done in documenting the extent to which we rely on semantic constraints in sentence comprehension.

3.2.5 Other factors

One additional effect documented by **Gibson and Warren (1998)** is that singly- and doubly-nested object-relatives receive lower complexity ratings when they use a first- or second-person pronoun, referred to as an *indexical* pronoun. In a questionnaire study, these were rated significantly less complex than sentences using a third-person pronoun, a short proper name, or a full noun phrase. Gibson (1998) accounts for this effect by assuming that the referents of *I* and *you* are always in the current discourse and thus can be accessed more easily. However, it may be possible to account for these effects simply by invoking the presumably high overall frequency of indexical pronouns in conversation, with the possible additional help provided by their short length.

3.2.6 Suggested experiments

This section suggests a few experiments that would be helpful in resolving open questions about the comprehension of relative clauses. Although these issues have not been considered worth investigating in the past because they were not crucial for distinguishing between sentence processing theories, they will presumably be of more interest once the CSCP model generates some predictions.

- It would be helpful to have a better controlled study of single center/right and subject/object-relative and passive embeddings. Although the frequency of passive relative clauses is 2-3 times that of object-relatives, no studies of passive RCs have been conducted. In order to control for semantic and pragmatic biases across clause types, sentences could be balanced in a design as follows:

  CO-1) The boy that the dog chased lost the cat.
  CO-2) The dog that the boy chased lost the cat.

  CS-1) The boy that chased the dog lost the cat.
  CS-2) The dog that chased the boy lost the cat.

  Together, the two CO sentences involve boy chasing dog, dog chasing boy, and both boy and dog losing the cat. Likewise, the two CS sentences involve the same four actions. Thus, semantic differences would be fairly well balanced across the sentence types. The inclusion of right-branching RCs as well would require 12 sentence types in each condition for complete counterbalancing of semantics.
- This study should be extended to include all pairs of nested relative clauses. In particular, experiments so far have not tested a subject-relative with another RC nested within it. A structure like this could be center-embedded if the subject-relative used a ditransitive and the first object were modified.
- The study should also be extended to include positive, neutral, and negative semantic biases, where sentences with negative biases contain plausible but unlikely relations. Proponents of constraint-based models, such as the CSCP, would hope to find significant effects of semantics.
- It would be interesting to examine the difference between using *who/which*, *that*, or no relative pronoun following human, animal, and inanimate nouns. Can comprehension difficulties in some of the experiments be explained by a confusing choice of pronouns? How much of an effect does reduction really have on each of the RC types? How well does frequency account for the effects?
- Experiments could also be performed with written and spoken input, where spoken input may or may not include natural prosodic cues. How do the different modalities compare? How much do listeners rely on prosody in spoken language?

3.3 Main verb/reduced-relative ambiguities

Researchers have long been interested in how we are able to comprehend temporarily ambiguous sentences and the conditions under which we are unable to do so. Although temporary ambiguities are very common, certain sentences can lead the listener to commit strongly to the wrong interpretation, only later revealing the correct analysis. In severe cases, the listener is unable to recover the correct interpretation. This is known as the *garden-path* effect (Ferriera & Clifton, 1986) and is exemplified by the now classic sentence from Bever (1970):

(23a) The horse raced past the barn fell.

This sentence is an example of the main verb/reduced relative (MV/RR) ambiguity, which has received considerable attention in the literature. The segment, "The horse raced past the barn," is typically interpreted as a complete sentence with *raced* serving as a main verb. But in this case, "raced past the barn," is in fact acting as a reduced relative, with *fell* playing the main verb. This ambiguity arises when there is a passive relative clause which is not introduced by a complementizer and auxiliary verb ("that was") and which uses a verb with identical past tense and past participle forms. The ambiguity does not work for the minority of verbs that retain different forms for the past tense (*ate*) and past participle (*eaten*).

Despite the potential for a garden-path effect, reduced relative clauses do not always result in noticeable comprehension difficulties. Consider, for example:

(23b) The hay carried past the barn had just been cut.

Hay could not plausibly carry something, except perhaps some needles, so mistaking *carried* to be a main verb is not as likely in this sentence. Furthermore, the verb *carried* occurs in the past participle form with equal or greater frequency than it does in the past tense form (MacDonald, Pearlmutter, & Seidenberg, 1994b). *Raced*, on the other hand, is used predominantly in the past tense and with an intransitive argument structure.

As with most ambiguities in natural language, a reader of the second sentence may be unaware that any ambiguity exists. **MacDonald, Pearlmutter, and Seidenberg (1994a)** and others have proposed several factors that may affect the chance that subjects are mislead by a MV/RR ambiguity. The following is a partial list:

1. The overall frequency (F) of main verbs versus reduced relatives, across the language.
2. The relative and overall F with which the ambiguous verb occurs in the past tense and past participle forms.
3. The F with which the verb uses a transitive argument structure.
4. The F with which the verb occurs in the passive voice.
5. The F with which the verb is used in a passive relative clause.
6. The F with which the passive relative clause is reduced, given the verb.
7. The animacy of the subject noun.
8. The semantic plausibility of the subject acting as either the agent or patient of the verb.

The most specific and reliable factor may be the probability that the sentence is completed as a MV or RR, given the noun phrase and the verb. Unfortunately, readers are unlikely to have received enough exposure to most noun/verb pairs to make this measure reliable, if one even had the capacity to store all of the data. Therefore, it is likely that skilled language users fall back on a combination of the more general variables or generalize from words with similar meaning.

The following sections discuss experimental findings relevant to some of these factors.

3.3.1 Animacy and semantic effects

Studies of how properties of the main noun affect the MV/RR ambiguity typically claim to be investigating either the animacy of the main noun or the semantic plausibility of the noun acting as either the agent or patient of the verb. But one should be careful in interpreting them, because the studies often do not distinguish plausibility from animacy or verb frequency effects. One could investigate the effects of plausibility in isolation by using either all animate or all inanimate nouns with the same verbs appearing in all conditions. But it is difficult to test effects of animacy in isolation from other semantic plausibility factors, although one could attempt this by collecting plausibility ratings and equating plausibility across animacy conditions.

In one of the earliest studies of plausibility effects on ambiguity resolution, **Rayner, Carlson, and Frazier (1983)** failed to find evidence that semantic plausibility has an immediate impact on reading time in MV/RR sentences. Based on the subjects' paraphrases, semantic biases did have significant effects on the eventual sentence interpretations, and positive biases improved overall reading times, but these effects did not show up in first-pass online reading times as measured by eye movements.

An analysis of the sentences used in this study indicates that the lack of online effects could be attributed to rather weak plausibility biases in the materials. Table 3.3 shows the initial segments of 6 of the 12 reduced sentence pairs used by Rayner et al. (1983). With the possible exception of "the bank," all main nouns were animate. In most cases, the intended MV-biased noun phrase appears to be, indeed, MV-biased. However, the biases of the intended RR-biased noun phrases are questionable. Is a *customer* less likely than a *publisher* to mail information? Teenagers are perhaps almost as likely to sell a car as they are to buy one.

It should also be noted that very few of the relative clauses used in this study were of the simpler direct-object extracted type ("The ball hit by the batter…"). Eight of the 12 sentences used nouns extracted from the indirect object of a verb of transmission ("The child read the story by her mother…," or, "The defendant denied the right of appeal…"). It is possible that these more complex relative clauses rendered the reduced relatives used in this experiment more difficult to comprehend than those typically examined in other studies, particularly since the length of the ambiguous region is drawn out.

Main verb biased	Reduced relative biased
the publishers mailed the information	the customers mailed the information
the hitchhikers expected to arrive	the visitors expected to arrive
the bank wired the money	the tourist wired the money
the customer paid the money	the clerk paid the money
the tourists asked for directions	the townspeople asked for directions
the dealer sold the car	the teenager sold the car

Table 3.3: Selected materials from Rayner, Carlson & Frazier (1983).

Pearlmutter and MacDonald (1992) and Tabossi, Spivey-Knowlton, McRae, and Tanenhaus (1994) noted that another potential problem with the materials in this study was that the semantic constraints were not simply a property of the noun/verb pair but of the noun, verb, and direct object of the verb. *Tourists* are not more likely to ask something than are *townspeople*, but may be more likely to ask for *directions*. Thus, the semantic constraints were not only weak but were late in arriving and were based not just on the pairing of noun and verb, but on the combination of noun, verb, and direct object.

The **Ferriera and Clifton (1986)** study provides stronger evidence against the immediate use of plausibility information in the MV/RR ambiguity. They, too, tracked eye movements during reading. Relative clauses were either reduced or marked and they modified either animate or inanimate nouns. All sentences were disambiguated by a prepositional phrase immediately following the ambiguous verb. Eye-tracked reading times were measured in the initial noun phrase, on the ambiguous verb, and on the prepositional phrase. Slightly longer reading times were found at the ambiguous verb when the noun was inanimate. But animacy was not found to interact with ambiguity resolution at the prepositional phrase. The effect of a reduced relative was virtually the same regardless of whether the noun was animate.

However, **Trueswell, Tanenhaus, and Garnsey (1994)** revisited the Ferriera and Clifton (1986) study and, with a few modifications, found quite different results. The most important change in methods was to strengthen the semantic context by eliminating inanimate noun/verb pairs that had plausible main verb interpretations. So it should be noted that this study is not investigating a purely animate/inanimate distinction, but a mixture of animacy and plausibility effects. The authors also included a condition with morphologically unambiguous verbs (*ate/eaten*).

Trueswell et al. found that subjects indeed had more difficulty reading reduced-relative sentences which had animate subjects. Reading times for reduced sentences with animate nouns, in comparison with inanimate nouns, were slightly faster on the verb (not significantly) but much slower on the disambiguating phrase. Reading times for sentences with inanimate nouns were no longer than for sentences with marked relative clauses. Second-pass reading times indicate that RRs with animate nouns were re-read far more often than either RRs with inanimate nouns or unreduced relatives.

The inanimate nouns in this study were chosen to be poor agents of the verbs, but were not necessarily good patients. So the authors reclassified the inanimate nouns based on whether they were judged to be good patients of the verbs, and thus supportive of the RR interpretation. Sentences with inanimate nouns that were poor agents and good patients of the verbs were no harder than sentences with unambiguous verbs. But inanimate nouns with weaker semantic fit incurred longer reading times. Regression analyses (Trueswell & Tanenhaus, 1994) confirmed a negative correlation between reading times and ratings of the typicality of a noun acting as the patient of its verb. The authors conclude that the semantic fit of a noun to potential argument positions of the verb can have immediate effects on resolving this ambiguity.

The results of the Trueswell et al. (1994) study have been replicated, and other semantic effects that do not rely on animacy have also been demonstrated. Five other studies found similar reading times between unambiguous sentences and ambiguous sentences with helpful context (Burgess, 1991; Pearlmutter & MacDonald, 1992; Tabossi et al., 1994; MacDonald, 1994; Burgess, Tanenhaus, & Hoffman, 1994). Other studies, reviewed in Section 3.7, revealed sensitivity to discourse context, prior to the ambiguous sentence, that affects the felicity of noun modification.

Pearlmutter and MacDonald (1992) separated plausibility from animacy by using only animate nouns. Nouns were rated by subjects as either good or poor patients of the ambiguous verb, and relative clauses were either reduced or marked. Self-paced moving window reading-time data showed that, at the ambiguous verb, plausibility had no effect on marked sentences. On reduced sentences, however, good-theme nouns led to significantly longer reading times than poor-theme nouns. This indicates that subjects were already entertaining the more difficult reduced-relative interpretation for the good-theme contexts. The opposite results were found at the point of disambiguation, with poor-theme nouns leading to much longer reading times as the subject attempts to recover from the incorrect MV interpretation. Pearlmutter and MacDonald (1992) also found in a regression analysis that the extent to which the nouns were rated by subjects as good themes of the verbs correlated with the reading times at the point of disambiguation such that better themes led to faster reading times.

Tabossi, Spivey-Knowlton, McRae, and Tanenhaus (1994) presented sentences with a two-word moving window. Nouns were either good agents or good patients of the verbs, and relative clauses were either reduced or marked. All of the good agents were animate, but some of the good patients were inanimate. In the *verb+by* region, there was little effect of semantics for reduced sentences. Marked sentences were faster in this region, and this effect was greater for good patients. In the following region, the NP within the by-phrase, there was a large slowdown for good agents and a smaller reduction effect. In the next region, reduced good subjects were very slow, marked good patients were fastest, and the other conditions were in the middle. The largest reduction effects (reduced sentences taking longer than marked sentences) occurred at the main verb for good agents, but occurred earlier for good patients. The point of maximum reduction effect was on the agent for good animate patients and even earlier, at the ambiguous verb, for inanimate patients. According to the authors, "[R]egression analyses provided evidence for the use of semantic information throughout the ambiguity resolution process." (p. 604)

The **Burgess, Tanenhaus, and Hoffman (1994)** study may help explain why some experiments have found that favorable context removes the reading time differences between MVs and RRs, while others have not. Previous work (Burgess & Tanenhaus, 1992), using word-by-word self-paced reading, found that, as in Ferriera and Clifton (1986), inanimate subjects failed to remove the garden-path effect. Burgess et al. (1994) used a two-word moving window in which the verb was presented along with the preposition *by*. The stimuli from Ferriera and Clifton (1986) again showed a garden-path effect, but more constraining material resulted in no effect on RR-biased nouns. Thus, the immediate availability of the preposition plays a role in alleviating the garden-path effect. The two-word presentation is presumably a bit more like natural reading, in which the preposition would be available parafoveally and would probably not be fixated.

McRae, Ferretti, and Amyote (1997) also used the two-word moving window method in investigating the MV/RR ambiguity. They contrasted sentences with identical nouns and verbs but with adjectives modifying the main noun that were meant to bias the interpretation toward either the MV or RR, as in (24a) and (24b), respectively. Sentences also appeared in marked form. Because the same nouns and verbs were used across conditions, animacy and verb specific effects were controlled for.

(24a) The shrewd heartless gambler manipulated by the dealer...

(24b) The young naive gambler manipulated by the dealer...

In the *verb+by* region, there was no effect of the noun bias and both were slower in the reduced form. However, in the next region, *the dealer*, there was no reduction effect for the patient-biased sentences, (24b), but a large effect for the agent biased sentences, (24a). In the next region there was again no effect of bias and only a small reduction effect. This finding is particularly interesting because it indicates that readers are sensitive to properties of the subject noun phrase that go beyond the particular lexical item used. That is, subjects are not basing parsing decisions on the lexical item, *gambler*, per se, but are forming a concept of a particular gambler, either naive or shrewd, and making rapid parsing decisions based on that newly constructed concept.

Finally, **Ni and Crain (1990)** showed that the felicity of noun modification affects RR interpretations on the MV/RR ambiguity. RRs with plural nouns preceded by *only*, which favors further modification, resulted in reading times similar to those for unambiguous sentences. On the other hand, the same nouns preceded by the definite article or an article and an adjective took significantly longer at the point of disambiguation. Other biases on the felicity of noun modification can be introduced by placing the sentence in a particular discourse context. Discourse effects are reviewed in Section 3.7.

3.3.2 Verb frequency effects

In addition to effects of semantic properties of noun/verb pairs, a number of studies have found effects of verb use frequency on the MV/RR ambiguity. The two primary factors of interest have been the frequency with which verbs appear with transitive argument structures and the frequency with which verbs appear in the past participle or past tense forms.

MacDonald (1994) investigated the role of alternate verb argument structures on the MV/RR ambiguity. Two sets of verbs were used: those that are always transitive and those that are optionally intransitive. Those that are always transitive were expected to support the reduced relative interpretation, especially when the presence of a direct object is ruled out. Two different types of 3-word phrases followed the verb. One set, the "good" post-ambiguity constraint, was immediately inconsistent with the active transitive (MV) interpretation. The "poor" post-ambiguity phrases were consistent with either interpretation for one or two words. Sentences were presented in a moving window, self-paced design.

MacDonald found that verbs with optional intransitive argument structures led to faster reading times on the verb and post-ambiguity phrase (the "reverse ambiguity" effect) but slower reading times on the disambiguating region than transitive-only verbs. The good post-ambiguity constraint had a greater effect on the optional verbs than it did on the transitive-only verbs.

MacDonald (1994) also combined an animacy manipulation with the post-ambiguity manipulation, using a mixture of transitive-only and optionally-transitive verbs. Compared with an unambiguous baseline, all conditions were slightly slower in the ambiguous region, but with both factors supporting a RR reading, there was no slow-down in the disambiguating and final regions. With only one of the factors supporting a RR, there was a small slowdown in these last two regions. But with both factors supporting a MV interpretation, reading was significantly slower. There was no early reverse ambiguity effect in these results, but that may be because the ambiguous and unambiguous sentences did not have the same number of words.

A third experiment compared optionally intransitive verbs biased toward the transitive structure with those biased toward the intransitive structure, with either good or poor post-ambiguity constraints. The comprehension data showed main effects of both verb bias and post-ambiguity constraint. Verbs biased toward the intransitive resulted in longer reading times at the disambiguation, but transitive-biased verbs did not. This indicates that the relevant factor is not necessarily the number of competing argument structures but the strength of the competing argument structures. This experiment also confirmed the reverse ambiguity effect on the verb and the word following it.

We should note that the MacDonald (1994) study, while manipulating verb argument structure, was not careful to control for other factors, such as the frequency of passives and past participles. Thus, more fully controlled studies may be necessary before we can definitively conclude that subjects are sensitive to the frequencies of verb argument structures in reading MV/RR sentences.

We turn now to studies of past participle frequency effects. Much of the evidence for these effects come from meta-analyses of previous studies that had manipulated other factors. **MacDonald, Pearlmutter, and Seidenberg (1994b)** reanalyzed the data from the Pearlmutter and MacDonald (1992) reading time experiment, which had directly addressed semantic plausibility. The reanalysis found that, among the set of higher plausibility items, verbs with more frequent past participle than past tense uses were associated with higher plausibility ratings.

MacDonald et al. (1994b) also reanalyzed the data from 12 previous studies of context effects in the MV/RR ambiguity. They found that the four studies that failed to show context effects used verbs with a lower (50% vs. 62%) relative past participle frequency than the eight studies that found significant context effects. This is suggestive, though not conclusive, evidence that past participle frequency played a role in producing those patterns of results.

Trueswell (1996) conducted a meta-analysis of the materials used in the Trueswell et al. (1994) study. He found that a verb's relative participle frequency had a significant negative correlation with reading time. However, these results are again only suggestive of an independent tense effect because semantic effects were not controlled for. Trueswell (1996) went on to show, using word-by-word self-paced reading, that reading times for reduced relatives with verbs having high relative past participle frequencies were no slower than reading times for marked relatives. Reading times for verbs with less frequent past participles were significantly higher in the disambiguating regions. All subject nouns in this study were chosen to be poor agents and good patients of the verbs, thus supporting the RR interpretation. An attempt was made to balance the semantic effects of the nouns between the high- and low-PP verb sets.

In a second experiment, nouns were chosen that supported the main verb interpretation. In this case, there was processing difficulty for both verb sets, although high-PP verbs led to faster reading times later in the sentence, following the disambiguating prepositional phrase. The authors suggest that this indicates the high-PP factor cannot by itself eliminate difficulty with reduced relatives, but it can have a significant effect in conjunction with a favorable semantic context.

While there is suggestive evidence of independent effects of tense morphology, argument structure, and semantic plausibility in resolving the MV/RR ambiguity, there are as yet no experiments explicitly manipulating more than one set of variables. And although voicing is postulated to play a role in lexical theories of syntactic ambiguity resolution (MacDonald et al., 1994a, 1994b), there are as yet no studies of the effect of the frequency with which a verb occurs as an active or a passive.

3.3.3 Reading span

Several studies have examined the effects of individual differences on the MV/RR ambiguity. One of the first of these was **Holmes (1987)**, who classified readers as good or average based on their ability to answer comprehension questions and as fast or slow based on reading speed. Only the good-slow readers were slower on relative clause sentences than on transitive ones and slower on reduced than on marked relatives. The other three groups showed no effect of sentence type on their reading speed. The good-slow and average-fast groups produced many more errors in comprehending reduced relatives. This indicates that there can be dramatic individual differences in reading ambiguous sentences. Holmes argued that intermediate readers, like the good-slow group, are the most common and thus dominate the effects in randomly sampled subject pools.

Just and Carpenter (1992) investigated the effects of animacy on MV/RR reading in Ss who were rated either high-span or low-span, based on their performance on the Daneman and Carpenter (1980) Reading Span task, which requires Ss to hold a series of words in memory while comprehending sentences. Using items similar to those used by Ferriera and Clifton (1986), but modified to strengthen the animacy effect, they found that only high-span Ss were sensitive to animacy in their by-phrase reading times. Both low-span and high-span readers showed a reduction effect, and the effect of animacy appeared independent of reduction in either group. Somewhat strangely, however, they did appear to find an effect of animacy on the RC verb which was *stronger* for the low-span readers than for the high-span readers. Furthermore, this effect was such that the verb was read 52 ms faster following inanimate subjects, which is opposite the result found in most other studies, including Ferriera and Clifton (1986). Although not discussed, this perplexing finding would seem to contradict their conclusion that, “Only the high span subjects use [animacy] in their first-pass reading.” (p. 128)

MacDonald, Just, and Carpenter (1992) also studied the effects of reading span on the MV/RR ambiguity but without manipulating context. While they found that high-span Ss were more accurate in responding to comprehension questions for RR sentences, the more interesting result was that high-span Ss had longer reading times on both RR and MV sentences. Two explanations of this result have been put forth. As suggested by the authors, it may be that, in the absence of strong contextual constraints, high-span Ss will entertain both the MV interpretation and the less frequent RR interpretation, even on MV sentences.

Another possibility is that the high-span Ss were sensitive to subtle differences in plausibility between the unambiguous and ambiguous MV sentences. Indeed, **Pearlmutter and MacDonald (1995)** showed that the average plausibility of ambiguous MV items in that study was lower than that of unambiguous MV items. The Ss may have been slower on the ambiguous MV sentences because of their lower plausibility, not because the Ss were maintaining the RR interpretation. Pearlmutter and MacDonald (1995) replicated the results of MacDonald et al. (1992) and found that high-span Ss were better able to use plausibility constraints to resolve MV/RR ambiguities in favor of the simpler MV interpretation and that their reading times at the point of disambiguation were correlated with the plausibility of the intransitive MV interpretation. The reading times of low-span Ss, on the other hand, were not correlated with plausibility ratings but were correlated with verb frequency information.

In an offline rating task, on the other hand, low-span Ss were equally sensitive to plausibility constraints. This suggests that low-span Ss did not lack the knowledge required to make use of plausibility constraints but lacked the ability to use that knowledge in an online fashion. The finding that reading times for high-span Ss on MV sentences were correlated with the plausibility of the intransitive interpretation was taken as support for the theory that the difference between high- and low-span Ss is not due to high-span Ss entertaining multiple interpretations but to their ability to make use of more subtle contextual constraints. However, the evidence does not seem to definitively rule out the possibility of multiple interpretations playing a role in slower high-span readings of MV sentences.

In the MacDonald (1994) study, reviewed above, ambiguous verbs were followed by phrases that disambiguated them as reduced relatives either immediately or after some delay. Across all Ss there was a small advantage in reading time for the early-disambiguating contexts. **MacDonald (1996)** reanalyzed this data based on the reading span of the Ss and found that the high-span Ss were solely responsible for the effect. The high-span Ss were significantly helped by the good context, reading these sentences almost as fast as unambiguous controls. Low-span Ss, on the other hand, were unaffected by the post-ambiguity context.

3.3.4 Summary

The studies of the MV/RR ambiguity reviewed here paint an overall picture in which readers are variably sensitive to a number of constraints at differing levels of complexity. At the simplest level, when a noun phrase (NP) is followed by a verb phrase (VP), there is a tremendously strong frequency bias in favor of the MV interpretation. Then there are statistical functions of the verb: the frequency with which it occurs as a past participle, with a transitive argument structure, or in passive voice. At the next level are noun/verb plausibility constraints: how likely is it that the noun can serve as the subject or object of the verb? There are also within-sentence pragmatic constraints affecting the likelihood that the subject NP is modified and post-ambiguity constraints that influence the likelihood that a particular verb form has occurred. Finally, to be discussed in Section 3.7, are discourse constraints affecting the felicity of NP modification.

There is mounting evidence that skilled readers can make use of each of these sources of evidence to guide their online reading. MacDonald (1994) demonstrated that Ss are sensitive to the frequency of verb argument structure. Meta-analyses (MacDonald et al., 1994b; Trueswell, 1996) and direct evidence (Trueswell, 1996) suggest that Ss are sensitive to the frequency of verb past participles. A number of the experiments have found evidence of plausibility effects either based on animacy (Trueswell et al., 1994; Tabossi et al., 1994; Just & Carpenter, 1992) or when controlling for animacy (Pearlmutter & MacDonald, 1992; McRae et al., 1997). Ni and Crain (1990) showed that Ss are sensitive to the felicity of noun modification, as determined by the article in the subject NP, and MacDonald (1994) found sensitivity to a post-ambiguity constraint that affects the likelihood that a transitive MV has occurred by ruling out the later appearance of a direct object.

Studies of individual differences on these tasks have consistently found that only high-span readers are sensitive to animacy, plausibility, and argument structure (Just & Carpenter, 1992; Pearlmutter & MacDonald, 1995; MacDonald, 1996). More work is needed to determine which effects were carried by high-span Ss in experiments that did not test reading span and what factors low-span Ss are actually sensitive to.

Although lexically-based sentence processing models, including connectionist models, propose that language users are able to combine constraints from multiple sources in guiding comprehension, there are relatively few experiments that actually manipulate multiple factors in order to determine their relationship. One such study is MacDonald (1994), which co-varied transitivity and post-ambiguity constraints in one experiment and transitivity and animacy in another. Both experiments found significant advantages in reading a RR when both factors supported the RR interpretation and disadvantages when both factors supported the MV. When only one factor supported the RR, reading times fell in the middle. However, these experiments do not provide any evidence that Ss were more sensitive to one factor over another.

The conventional wisdom seems to be that biasing constraints have their greatest influence when the two possible interpretations, MV and RR in this case, are equally likely. If one interpretation is dominant, a biasing constraint will have less effect. This can help to explain the contradictory results seen in much of this literature. For example, some of the experiments that managed to find effects of verb frequency, where others did not, used nouns that were poor agents and good patients of the verbs to reduce the MV bias, resulting in greater sensitivity to verb frequency. To better understand how constraints interact, we need more experiments that vary multiple factors simultaneously.

One final open issue is that of early reverse ambiguity effects. MacDonald (1994) found that, in contexts that were biased in favor of the RR interpretation, Ss were faster at the point of disambiguation but *slower* on and just following the ambiguous verb. This effect was confirmed by two other studies as well (Pearlmutter & MacDonald, 1992; McRae, Spivey-Knowlton, & Tanenhaus, 1998). That we should find a reverse ambiguity certainly makes some sense. A RR sentence is presumably more complex than a simple MV sentence. When a reader encounters the verb in a strongly biased RR context, she may immediately recognize it as a RR and incur a processing cost in preparation of encoding the RR. In the absence of a strong bias in favor of a RR, this cost occurs later, at which point more effort is required to transform the MV representation that has formed. However, three studies failed to find such reverse ambiguity effects (Trueswell et al., 1994; Tabossi et al., 1994; Trueswell, 1996). It remains to be seen under what circumstances a reverse ambiguity effect can be reliably demonstrated.

3.3.5 Suggested experiments

- As mentioned in the summary, while the frequency with which a verb occurs in the passive is thought to play a role in the MV/RR ambiguity, no studies of this factor have been conducted as yet.
- To really understand which verb-related factors are useful to readers and how they interact, we need a study involving a large number of verbs which have been scored for transitivity, passivity, past participle frequency, and RC reduction frequency. Hopefully, enough verbs can be found so that the effects of individual factors can be isolated. Ideally, this would also cross a semantic factor with the verb-specific ones and compare high- and low-span readers.

  While this design may be too ambitious, there does not seem to be any better way to understand the role played by each of the factors thought to be relevant to the MV/RR ambiguity. If some factors are not controlled for, we may just obtain more misleading results. This experiment would hopefully shed more light on the conditions under which the reverse ambiguity effect occurs.

3.4 Sentential complements

Another interesting class of ambiguity in sentences is that caused by a reduced sentential complement (SC). A sentential complement is a sentence that acts as a verb or noun complement. In (25a), "[that] pigs can fly" is a sentence acting as an object of the verb *believe*. In (25b), the same sentence acts as a complement of the noun *belief*. Most of the work on SCs deals with the former, verb-complement, type and this review focuses on experiments using that form.

(25a) I believe [that] pigs can fly.

(25b) The belief [that] pigs can fly is sorely mistaken.

However, it is worth noting that Gibson and Thomas (1997) have recently published interesting results on the comprehensibility of noun-complement SCs in conjunction with relative clauses. Subjects rate a sentence, such as (26a), which has an RC within an SC, as significantly more comprehensible than one with an SC within an RC, such as (26b). Because such nested structures are extremely rare, if they occur at all in actual usage, it would not be practical to include them in the language on which the current model is trained. As with any structure that is not regularly used in a language, it is not clear whether sentences such as (26b) are not used because they are incomprehensible or are incomprehensible because they are rarely or never encountered.

(26a) The fact that the employee who the manager hired stole office supplies worried the executive.

(26b) # The executive who the fact that the employee stole office supplies worried hired the manager.

Evidence to help in resolving such a debate could be obtained by training a comprehension network on a language in which the two structures occur equally often. If the network has more trouble with the structure that is precluded in English, it suggests that the structure is incomprehensible because it intrinsically is difficult for a connectionist parser to handle (or a parser trained on an otherwise representative subset of English) and not simply because it is rare. Such an experiment, while possible using the current model, has yet to be attempted.

Verb-complement SCs, as in (25a), can only follow certain verbs, which are usually verbs of thinking or communicating. These verbs can often take a noun complement as well, and when such a verb is followed by a reduced SC, there is a temporary ambiguity. The noun phrase following the verb could be its object or it could be the subject of an SC. Because non-sentential objects are so much more common in the language overall, these are typically considered the preferred reading, and are the reading that the theory of *minimal attachment* (Frazier, 1979, 1987) suggests will always be the first interpretation because its structure requires the fewest parse tree nodes. Although, as we will see, it is clear that the difficulty of a reduced SC depends on the bias of the verb, and possibly on the noun phrase as well, at issue is whether the noun phrase complement (NP) is always the first interpretation considered.

Constraint-based theories hold that the minimally attached hypothesis is not necessarily the first one entertained. They predict that subjects should be sensitive to a variety of factors, including the frequency of the verb taking an SC and the probability that the SC is reduced, as well as semantic and context effects, in forming their initial reading. To rule out the minimal attachment hypothesis, it must be demonstrated that strong constraints can eliminate any suggestion of an initial minimal attachment interpretation.

Mitchell and Holmes (1985) tested phrase-by-phrase reading times of SC sentences like (27) using verbs that favored either an NP or an SC. When the SC was reduced (didn't include the complementizer *that*), NP-biased verbs incurred much longer reading times at the disambiguation point than did SC-biased verbs. This difference was not found with marked SCs. For NP-biased verbs, a complementizer appeared to speed up reading times, but for SC-biased verbs, the complementizer had the opposite effect. Although this shows sensitivity to the frequency with which a verb is followed by an SC, it does not indicate whether the NP was actually the preferred initial reading.

(27) The historian suspected [that] the manuscript of his book had been lost.

Holmes, Kennedy, and Murray (1987) compared sentences with reduced and marked sentential complements to those with NP complements. They found that cumulative self-paced, word-by-word reading times in the disambiguating region were similar for reduced and marked SCs but were shorter for NPs. The fact that an overt complementizer did not improve reading times was taken as evidence against a minimal attachment strategy.

Because the disambiguating phrases were different between the SC and NP sentences, it is unclear what led to the longer reading times for NPs. It could have been due to an initial interpretation of an NP, even for the unambiguous SC. Or, more likely, it was due to the fact that the disambiguating phrases for the SC sentences were more complex verb phrases and thus required more processing in their own right. A valid direct comparison between SC and NP reading times may not be possible because they require different disambiguating phrases. Thus, that an initial NP reading always occurs with a reduced SC may have to be established by showing that reduced SCs always incur longer reading times than marked SCs.

Rayner and Frazier (1987) used similar, though not identical, sentences to Holmes et al. (1987) but employed the arguably better eye-movement measure, rather than self-paced reading. They found longer reading times in the disambiguating region for the reduced sentences than for either of the other two sentence types, thus supporting the minimal attachment theory. Again, because the disambiguating regions differed, it is not clear what is behind these results.

However, **Holmes (1987)** did a post-hoc partitioning of the verbs in the Holmes et al. (1987) study into NP- and SC-biased sets. Only the reduced sentences with NP-biased verbs took longer to read than the marked sentences at the disambiguation. For SC-biased verbs, the effect was actually in the opposite direction, although not significantly so. Holmes (1987) then conducted a second experiment that explicitly manipulated verb bias. This resulted in a very large reduction effect for NP-biased verbs and only a small reduction effect for SC-biased verbs.

Holmes, Stowe, and Cupples (1989) attempted to clarify these findings. They again compared SC- and NP-biased verbs with reduced and marked SCs. They also varied the plausibility of the NP following the verb to be either a good or a poor possible theme of the verb, based on subject ratings. Sentence presentation was self-paced and cumulative (words remained visible) and subjects performed a continual grammaticality decision. The only effect of NP plausibility was a slight increase in reading time at the noun for NP-biased verbs with poor-themed nouns and significantly more grammaticality decision errors made for reduced NP-biased sentences with good-theme nouns. For NP-biased verbs, reduced sentences took much longer to read at the start of the disambiguating verb phrase, following the noun. For SC-biased verbs, there were small but significant increases in reading time for reduced sentences at the determiner of the noun phrase and the start of the disambiguating verb phrase.

However, these results are questionable because the continual grammaticality decision task and the cumulative presentation may have affected how subjects were reading the sentences. So Holmes et al. (1989) redid the experiment asking subjects to try to remember and reproduce the sentences. In this case, there was a significant reduction effect for NP-biased verbs in the disambiguating verb phrase, but there were no effects of NP plausibility or reduction for

SC-biased verbs. It is apparent that readers are sensitive to verb bias in reading ambiguous SC sentences and that strongly biased verbs can result in similar reading times in reduced and marked sentences.

A third experiment added longer noun phrases (containing a prepositional phrase) and also changed to non-cumulative presentation and question answering rather than sentence repeating. Both the short and the long NPs showed a reduction effect for NP-biased verbs but only the long NPs showed a reduction effect for SC-biased verbs. Therefore, the authors were tempted to conclude that a longer noun phrase leads readers to expect an NP, rather than an SC. It is not clear how this result should be interpreted. It may be that SCs rarely have long, modified subjects or it may be that SCs with long subjects are rarely reduced. If either is the case, subjects could be relying on this subtle form of constraint. The fact that a stronger reduction effect is evident for long NPs should not necessarily be taken as evidence that subjects are entertaining an initial minimal attachment hypothesis. And complicating all of this is the fact that the longest reading times at disambiguation actually occurred for the reduced, NP-biased condition with *short* NPs.

Interestingly, **Ferreira and Henderson (1990)** found quite different results in a study of the SC ambiguity. In one experiment, they used eye-tracking to measure reading times on SC sentences with SC- and NP-biased verbs and with and without complementizers. First-pass reading times were longer for reduced sentences in the disambiguating region for both sets of verbs. A second experiment tested the same sentences using a one-word self-paced moving window. In this case reading times were longer at the disambiguating and post-disambiguating regions for reduced SCs, but there were no significant effects of verb type. However, in this case there was a trend toward faster reading times for SC-biased verbs. In a final experiment, cumulative displays were used, as in Holmes et al. (1987) and the first two experiments of Holmes et al. (1989), rather than the moving window. In this case effects were weaker overall and there was only a marginal effect of reduction at the disambiguating region.

The findings of Ferreira and Henderson (1990) are a bit hard to reconcile with other experiments clearly demonstrating sensitivity to verb biases (Mitchell & Holmes, 1985; Holmes, 1987; Holmes et al., 1989; Trueswell, Tanenhaus, & Kello, 1993). There are some possible problems with the materials used in this experiment that may be responsible for these differences. Some of the sentences were rather awkward or did not quite make sense. Because there were more test items than distractors, there was a very high proportion of sentential complements, which could have affected subjects' readings. Furthermore, the sentences were all quite short and subjects were only periodically asked comprehension questions, which they almost always got correct. In contrast, other experiments involving comprehension questions rarely report greater than 90% correct answers. This suggests that the questions may have been too easy, which would lessen the pressure to read carefully.

More fundamentally, as Trueswell et al. (1993) pointed out, the verb bias manipulation used by Ferreira and Henderson (1990) was relatively weak. Based on my own analysis of the Wall Street Journal and Brown Corpora in the Penn Treebank (see Section 4.4), this seems to be the case. The SC-verbs and the NP-verbs were fairly well matched for overall frequency, with the SC-verbs having a slightly higher mean frequency (SC:70.0 NP:67.6) and the NP-verbs having a higher mean log frequency (SC:3.5 NP:3.8). However, the verbs were also fairly *well* matched for frequency of occurrence with a sentential complement (SC:19.5 NP:16.8) and for the chance that they are followed by an SC (SC:33.9% NP:29.9%). Although the advantage is in favor of the SC-verbs, it is not much of an advantage. The SC-verbs *were* more likely to have a reduced SC (SC:35% NP:16%), but the frequency of taking a reduced SC is quite low for both of these verb sets, relative to verbs more commonly occurring with an SC.

Trueswell, Tanenhaus, and Kello (1993) raised the question of whether longer reading times for reduced SCs necessarily indicate syntactic misanalysis. Longer times could simply reflect the cost of representing multiple parses in parallel. The researchers therefore attempted to find a complementizer effect that could be separated from syntactic misanalysis. In the first experiment, subjects listened to a sentence fragment and then immediately read a word out loud. There is good evidence that words that are valid continuations of the sentence should be read faster than words that are not. Sentence fragments consisted of an NP followed by a verb and then an optional *that*. The verbs were either SC- or NP-biased. The visually presented word was either a pronoun in the accusative case (*him*), which is inconsistent with the SC reading, or in the nominative case (*he*), which is inconsistent with the NP reading. Subjects also judged whether the word was a good continuation of the sentence.

When a complementizer was present, *him* took longer to read than *he*, with no effect of verb type, indicating that subjects were sensitive to the fact that *him* cannot begin an SC. When the complementizer was absent, *him* took significantly longer for SC-biased verbs than for NP-biased verbs and *he* took non-significantly longer for NP-biased verbs. This indicates that, in cases of ambiguity, subjects are sensitive to verb type. Reading times for *him* were longer with the complementizer present than with it absent only for NP-biased verbs, which supports the conclusion that subjects did not entertain an initial NP reading following the SC-biased verbs.

Naming times for *he* were longer for SC-biased verbs without a complementizer than with a complementizer. The authors argue that this cannot be attributed to misparsing because subcategorization information was clearly available to and used by subjects based on other comparisons and because *he* is clearly case marked. They suggest this effect may be due to a general preference for the presence of a complementizer and that, in its absence, the process of building the SC structures is concentrated at the noun. The former hypothesis is supported by the fact that the verbs' complementizer preference was highly correlated with the complementizer effect in this experiment.

In a second experiment, Trueswell et al. (1993) tested self-paced reading times on SC sentences, such as (28), with and without complementizers and with NP- and SC-biased verbs. In comparison with the Ferreira and Henderson (1990) study, the sentences here used strongly biased verbs, noun phrases that were plausible objects of the NP-biased verbs, and more distractor items. Noun phrases also contained two words, rather than one word, reducing the chance that reactions to the noun phrase are delayed until the following verb.

(28) The student forgot/hoped [that] the solution was in the back of the book.

At the determiner (*the*) following the verb or complementizer, there was an increase in reading time for reduced sentences with a greater increase for SC-biased verbs. At the noun (*solution*) there was an elevated reading time only for the reduced SC-biased sentences. At the auxiliary verb (*was*) there were no effects, but at the next word (*in*), there was a much longer reading time for reduced sentences but only for NP-biased verbs. These results suggest that subjects were only garden-pathed by NP-biased verbs without a complementizer. There were early elevated reading times for the reduced SC-biased sentences, probably not because the subjects had misparsed the sentences but because of the cost of constructing the SC parsing structures. In short, subjects were clearly sensitive to the verb bias.

In a final experiment, the same materials were used in an eye-tracking paradigm. There were no significant first-pass reading time effects at the verb and noun phrase. At the disambiguating verb phrase, NP-biased verbs resulted in a significantly longer reading time without a complementizer, indicating the reduction effect. SC-biased verbs took a bit longer at this point when they were reduced than marked, but the effect was not significant. Again, there was a significant correlation between reduction effect and verb complementizer preference. However, in this case there was not an elevated reading time for reduced sentences with SC-biased verbs at the NP region, as there was in the self-paced reading. The authors attributed this difference to preview effects common to short function words. When the complementizer was present, subjects presumably skipped it and thus spent more time on the following NP. Trueswell et al. (1993) included a detailed analysis of fixation durations and probabilities that helps explain some of the differences between the results in their study and those of Ferreira and Henderson (1990).

The authors draw the following conclusions: Verb subcategorization information is available almost immediately and can affect the syntactic analysis of an ambiguous NP. SC-biased verbs can eliminate misanalysis of reduced SCs. Also, there is a processing difficulty for reduced SCs that is not associated with syntactic misanalysis, and this effect is correlated with the degree to which a verb expects a complementizer with its SC.

Juliano and Tanenhaus (1993) replicated some of the results from Trueswell et al. (1993). In one experiment they tested self-paced reading time on sentences with sentential complements or sentential subjects which began with *that* followed by an adjective, a noun, and the verb phrase. The noun was either singular or plural. Singular nouns (*that cheap hotel*) can support the interpretation that the SC is actually just an NP with *that* as a determiner, but plural nouns do not support this interpretation. Juliano and Tanenhaus found that Ss had longer reading times for verbs following singular rather than plural nouns in sentential complements. But in sentential subjects, there were longer reading times for the verb following plural nouns. Thus, readers appear to interpret a sentence initial that-phrase as an NP with *that* as a determiner, but they interpret a that-phrase following a verb as the start of an SC. The authors attribute this to readers' sensitivity to the prevailing statistics of that-phrases in English.

In two other experiments, Juliano and Tanenhaus (1993) used self-paced reading to study SCs starting with the determiners *that*, *the*, or *those*. *That* is ambiguous as a possible complementizer. Verbs were either SC-biased or NP-only and the results showed that readers were sensitive to the verb bias. SC-biased verbs were actually followed by reduced SCs, and NP-only verbs were actually followed by NPs. Any uses of *that* were actually determiners, not complementizers. Following SC-biased verbs, there was longer reading time at or just following the point at which *that* was disambiguated as a determiner than for the same regions following *those* or *the*. This suggests that *that* was initially interpreted as a complementizer following SC-biased verbs. Following NP-only verbs, there were longer reading times for *that* immediately and on the following words. This suggests that subjects were sensitive to the conflict between the possible roles of *that* as a complementizer and as a determiner, even when it could not, grammatically, have been a complementizer.

The authors also showed that reading times for *the* following an SC-biased verb were correlated with the verb's that-preference. For high-frequency verbs (which tend to have a low that-preference) reading times for *the* were similar for SC-biased and NP-biased verbs. This suggests that, when a complementizer is not expected, an SC reading following an SC-biased verb can be as easy as an NP reading following an NP-biased verb. Juliano and Tanenhaus (1994) constructed a simple recurrent network to predict complement type given the verb and the following word. The model was trained on frequencies drawn from the Treebank Corpus and produced a similar frequency by that-preference interaction.

Garnsey, Pearlmutter, Myers, and Lotocky (1997) argued that object plausibility was confounded with verb preference in the Trueswell et al. (1993) materials. A rating study showed that NPs following the SC-biased verbs were less plausible as direct objects than were NPs following NP-biased verbs. Therefore, Garnsey et al. (1997) designed an experiment to distinguish verb bias effects from plausibility effects. Based on sentence completions, three verb classes were constructed: SC-biased, NP-biased, and equally biased (EQ-biased).[4] Rating studies were also conducted to determine the degree to which nouns were good direct objects or good subjects in SCs of the verbs. For each verb, two nouns were chosen, a good object and a poor object, with the plausibilities fairly balanced across verb classes.

Using eye-tracking, Garnsey et al. (1997) were able to generally replicate the verb bias effects of Trueswell et al. (1993) and found that they could not be attributed solely to plausibility. SC-biased verbs caused no problems at the disambiguation, but NP-biased verbs did, and neither of these classes was significantly affected by plausibility. However, as expected, plausibility did significantly affect the EQ-biased verbs. There was actually a reverse ambiguity effect for EQ-biased verbs with implausible objects, which is a bit hard to explain. Unlike Trueswell et al. (1993) and Juliano and Tanenhaus (1993), there was no significant correlation between the SC-biased verbs' that-preference and slowed reading at either the NP or disambiguating region.

A second experiment used the same materials in the self-paced moving window paradigm. There were again significant problems at the disambiguation only for NP-biased verbs and for EQ-biased verbs with plausible objects. The reverse ambiguity effect for EQ-biased verbs with implausible objects did not replicate. In both experiments, *that* was read significantly faster following NP-biased verbs than following SC-biased verbs, with EQ-biased verbs falling in the middle. Thus, verb bias was made available fast enough to affect processing on the very next word. Although it is not entirely clear whether the effects of verb bias or semantic plausibility had more influence in this experiment, Garnsey et al. argue that verb information is likely to be more reliable and available.

In contrast with Garnsey et al. (1997), **Pickering, Traxler, and Crocker (2000)** found positive effects of plausibility even with SC-biased verbs. They used eye tracking to measure the reading time of reduced SC sentences such as (29a) and (29b). Based on a completion study, main verbs were strongly SC-biased but permitted direct objects. Sentence pairs differed only in whether the subject of the SC was a plausible (*potential*) or implausible (*exercises*) object of the verb. Although the experimenters found no plausibility effects in first-pass reading times, subjects were faster for plausible objects in the post-noun section (*one day*) in measures of right-bounded time, regression-path time, and total reading time.[5] This suggests that subjects were more likely to adopt the NP reading for plausible objects than they were for implausible objects.

(29a) The young athlete realized her potential one day might make her a world-class sprinter.

(29b) The young athlete realized her exercises one day might make her a world-class sprinter.

In the disambiguating region (*might*) subjects were slower, by right-bounded time only, for plausible objects. This presumably reflects recovering from the NP garden-path. A second experiment placed the sentences inside a short passage. The bias of the context for or against a particular interpretation was not carefully controlled. The pattern of results was similar, though not identical to those in the first experiment. Subjects were faster in the post-noun region for plausible objects based on first-pass, right-bounded, and regression-path reading times. However, there were no

[4] A potential problem with using sentence completions as indicators of the degree to which a verb prefers a sentential complement is that NP completions are typically shorter and thus quicker to write. Lazy subjects might skew the results toward non-SC completions.

[5] *Right-bounded time* is the sum of all fixations on the region of interest until the subject moves to the right of the region. *Regression-path time* is the time spent between first fixating a region and fixating to the right of the region. This includes fixations back to prior regions.

significant effects on the disambiguating region.

Overall, this experiment indicates that subjects are immediately sensitive to plausibility even for SC-biased verbs. This is in contrast with the results of Garnsey et al. (1997), who found plausibility effects only for EQ-biased verbs. Pickering et al. (2000) criticized the Garnsey et al. (1997) experiment on the grounds that it lacked enough data for good statistical power. The experiments also differed in that Garnsey et al. (1997) first measured reduction effects (differences in reading time with and without the complementizer) and then compared these across plausibility conditions. Pickering et al. (2000), on the other hand, included only reduced SCs and directly compared their reading times for plausible and implausible nouns. This is possibly a more sensitive measure, as plausibility could potentially affect both reduced and marked interpretations. However, looking at the data, it appears doubtful that performing this analysis on the Garnsey et al. (1997) would result in a plausibility effect for SC-biased verbs.

3.4.1 Summary

Research on the sentential complement, or NP/S, ambiguity has focused on three main factors: whether the SC is reduced, whether the main verb prefers an SC or NP complement, and whether the noun phrase following the verb is a good theme of the verb or a good subject of an SC. Because of their strong interaction, it is difficult to talk about the effects of reduction in isolation from the effects of verb preference.

One finding that is consistent across a number of experiments is that reduced SCs cause longer reading times at the point of disambiguation than marked SCs if the verb prefers an NP complement, but not if the verb prefers an SC complement. In other words, there is a reduction effect only for NP-biased verbs. This was the pattern found in Mitchell and Holmes (1985), in Experiments 2 and 3 of the Trueswell et al. (1993) study, in the reanalysis of the data from Holmes et al. (1987) conducted by Holmes (1987), and in the overall reading times in Garnsey et al. (1997). This was essentially the result in Experiment 1 of Holmes et al. (1989), although they did find a small but significant reduction effect for SC-biased verbs. However, the pattern again emerged in Experiment 2 and for the short NPs in Experiment 3 of that study. It should be noted that this result was not achieved by Ferreira and Henderson (1990), but the strength of the manipulation in that experiment has been criticized, as discussed above. Whether SC-biased verbs show a small reduction effect, no reduction effect, or a reverse reduction effect (faster reading times for reduced SCs) is inconsistent across experiments and may depend on the strength of the bias.

A related but somewhat different general finding is that verb preference has an effect for reduced but not for marked SCs. This pattern was found at the disambiguation point by Mitchell and Holmes (1985) and at the start of the SC in Experiment 1 and in first-pass reading times at the disambiguation point in Experiments 2 and 3 of Trueswell et al. (1993). Again, this finding was not supported by Ferreira and Henderson (1990). If accepted, these findings together suggest that there is only a garden-path effect at the point of disambiguation when a *reduced* SC follows an *NP-biased* verb.

Noun semantics, or the plausibility that the noun introducing the SC is either a good object of the verb or a good subject of an SC, appears to play a lesser role than verb preference. Holmes et al. (1989) found only a small effect of plausibility for NP-biased verbs. In most of their conditions, however, there was no effect of plausibility. Garnsey et al. (1997) found an effect of plausibility, but only for equally biased verbs that did not strongly favor an SC or NP complement. Pickering et al. (2000), on the other hand, did find immediate effects of plausibility for SC-biased verbs. It appears that readers are probably sensitive to plausibility, but its effect is not as strong as that of verb preference.

A final issue to consider is the possible presence of an early reverse ambiguity effect, similar to that found by MacDonald (1994) and others for the MV/RR ambiguity. The theory is that subjects will slow down at the start of an ambiguous phrase if they are interpreting that phrase not as an NP, but as a more complex SC. In their second experiment, Trueswell et al. (1993) found elevated reading times at the determiner and noun phrase for reduced SC-biased verbs relative to marked sentences or those with NP-biased verbs. However, this pattern did not appear in their third experiment. Juliano and Tanenhaus (1993) found longer reading times for *that* following SC-biased verbs, which could be attributed to a reverse ambiguity effect.

However, several other reading time experiments failed to find an early reverse ambiguity effect (Holmes et al., 1989; Ferreira & Henderson, 1990). The data from Garnsey et al. (1997) showed a possibly non-significant reverse ambiguity effect for SC-biased verbs followed by nouns that were plausible objects, but not with implausible objects. If confirmed, this would seem to go against the theory that the reverse ambiguity effect reflects early construction of an SC reading, as that reading should be more likely with an implausible object. It is possible that there are actually multiple components to the reverse ambiguity, with one reflecting expectation of an SC and another reflecting a possibly simultaneous difficulty with constructing an NP reading following SC-biased verbs. More research is necessary to confirm the existence of an early reverse ambiguity effect in the SC ambiguity and to determine how it relates to verb-bias and plausibility factors.

3.5 Subordinate clauses

Another type of ambiguity that has received some attention in the literature occurs when a subordinate clause at the start of a sentence contains a verb that can be either intransitive or transitive. Examples of this ambiguity, which is also known as the NP/0 ambiguity, are shown in sentences (30a) and (30b).

(30a) After the private had fainted [,] the sergeant decided to end the military drill.

(30b) After the private had saluted [,] the sergeant decided to end the military drill.

Frazier and Rayner (1982) studied this ambiguity in a reading-time experiment involving eye-tracking. Their design involved two factors: whether the ambiguous NP was an object of the verb (*late closure*) or the subject of the main clause (*early closure*) and whether it was short (*a mile*) or long (*a mile and a half*). Overall, they found longer reading times in the disambiguation region for early closure sentences, though this effect was more pronounced in second-pass reading times. This result is somewhat hard to interpret, however, because the disambiguation region differed in the two conditions. An additional finding was that the slowdown at the disambiguating region for early closure sentences was greater when the ambiguous noun phrase was longer. This suggests that the longer a reader is committed to one reading, the harder it is to change to a different analysis. Frazier and Rayner also argue that these results indicate that subjects initially assign the late closure reading and only abandon it when it proves untenable.

However, an additional factor, not addressed by Frazier and Rayner (1982), which may be relevant to the processing of this ambiguity, is the argument-structure preference of the verb. If the comma is omitted, one might expect that (30b) would cause more difficulty than (30a) because *saluted* frequently takes a direct object while *fainted* never does. **Mitchell and Holmes (1985)** confirmed this in two phrase-by-phrase reading experiments. In both cases, reading times for the disambiguating region (*decided to end*) were longer with NP-biased verbs, like *saluted*. This effect became non-significant when the comma was included in the display. This suggests that readers are sensitive to verb bias in the probability or strength with which they adopt the late closure reading.

Using phrase-by-phrase materials with a single break following the ambiguous NP (*the sergeant*), **Mitchell (1987)** reconfirmed that intransitives, in comparison to optionally transitives, result in shorter reading times on the second half of the display. However, intransitives also resulted in longer reading times on the first half of the display. According to Mitchell (1994), Adams, Clifton, and Mitchell (1991) confirmed this result using eye-tracking in an unsegmented display. Mitchell suggests that this indicates the verb information was not being used immediately because, had the information that this was an intransitive verb been available, the NP could have been immediately interpreted as the subject of the matrix clause, and would thus not have resulted in a delay.

On the contrary, this data seems to support the immediate use of the verb argument structure. Had the argument structure not been used, there should have been no difference between the verb types. The longer reading time for *the sergeant* following *fainted* presumably reflects the effort of constructing the matrix clause parsing structures. This is not done following *saluted the sergeant* because it is presumed that the matrix clause has not yet begun. Thus, the slowdown may be partially attributable to the preparation for future structure. Mitchell (1987) reported that earlier work failed to find these delays in conditions using optionally transitive verbs with commas present, where one should expect the reader to prepare for a new clause. However, as we'll see shortly, Sturt, Pickering, and Crocker (1999) did find immediate slowdowns associated with a comma in such sentences. Thus, longer delays in the intransitive case may be partially due to wrap-up of the subordinate clause or preparation for the main clause. It is also possible that these delays reflect competition between the argument structure preference of the verb and the overall tendency to interpret an NVN construction transitively. Although this competition may not be resolved instantly, the verb argument structure preference may begin asserting its effect immediately.

Ferreira and Henderson (1991) conducted a series of experiments on the subordinate clause ambiguity using grammaticality decision as a measure of comprehension difficulty. In the first two experiments, they compared early and late closure sentences in which the ambiguous region was either a simple NP, an NP with a marked object-relative, or an NP with a reduced object-relative (OR). All but the second experiment involved rapid one-word-at-a-time presentation of the sentences, with each word visible for 250 ms. The second experiment used phrase-by-phrase self-paced reading. Sentences with short ambiguous regions were judged to be fairly grammatical in both early and late closure conditions, while the reduced OR conditions were both judged to be ungrammatical. The marked OR, on the other hand, was judged to be ungrammatical only in the early closure condition. The authors concluded that increasing the length of the critical regions increases the difficulty of such sentences, with a greater effect on the early closure sentences. The next question is whether this is really due to length itself or to increased syntactic complexity.

In the third experiment, Ferreira and Henderson (1991) replaced the reduced OR condition with a prepositional phrase (PP) that matched the marked OR condition in terms of length. They found that both the PP and marked OR sentences were harder in the early closure condition than in the late, but there was no difference between them. They conclude, under the assumption that PPs and ORs differ in syntactic complexity, that syntactic complexity does not affect the difficulty of reanalysis. In experiment four, the PP condition was replaced with one in which the ambiguous NP has prenominal adjectives (*the big and hairy dog*). In this case, the prenominal adjectives were nearly as easy as the short NPs and were easier than the marked OR, especially in the early closure case. They interpret this to mean that the relevant issue is not length itself but, rather, the distance between the ambiguous noun and the point of disambiguation.

This was supported by the results of a fifth experiment, in which the marked OR was replaced by a predicate adjective RC (*the dog that is hairy*). Although presumably of similar semantic complexity, this new condition was judged harder than the prenominal adjective case. However, Sturt et al. (1999) failed to replicate the head-position effect using an online reading-time measure. Ferreira and Henderson (1993) did replicate the head-position effect using grammaticality judgments following eye-tracked reading, and it showed up in total reading times, but not in first-pass reading times.

Sturt, Pickering, and Crocker (1999) reported two self-paced reading time experiments comparing the subordinate clause ambiguity with the reduced sentential complement ambiguity. Subordinate clause sentences, as in (31a) and (31b) were all of the early closure variety and appeared with or without the disambiguating comma. The sentential complement sentences were either marked, (31c), or reduced, (31d). All verbs were biased toward the transitive, NP object, reading. Presentation was by blocked regions, rather than word-by-word. The vertical bars in the example sentences separate the regions.

(31a) Before the woman | visited, the famous doctor | had been drinking | quite a lot.

(31b) Before the woman | visited the famous doctor | had been drinking | quite a lot.

(31c) The Australian woman | saw that the famous doctor | had been drinking | quite a lot.

(31d) The Australian woman | saw the famous doctor | had been drinking | quite a lot.

Sturt et al. found that the second regions were read slower in the unambiguous sentences, with a nonsignificant tendency for the subordinate clause sentences to be slower. However, care should be taken in interpreting this result because the words in this region differed between conditions. The third region, which was identical across the conditions, is the more interesting one. Here there were main effects of both ambiguity and construction type. Ambiguous sentences were slower, as were subordinate clause sentences. The effect of ambiguity was stronger in the subordinate clause case. This seems to indicate that either misanalysis is more common for ambiguous subordinate clause sentences or is harder to recover from when it happens, or both. There are two possible explanations for this that do not involve any assumptions about either syntactic or semantic differences between the two types of ambiguity. The first is that reduced sentential complements are far more common than ambiguous subordinate clauses, which really only occur in spoken sentences that lack any prosodic marking of the clause boundary. The second explanation is that the missing comma in (31b) is really a grammatical violation, especially in a context in which the commas are used at least half of the time. In contrast, the reduced SC in (31d) is perfectly natural in English. The second experiment in this study was a failed replication of the head position effect of Ferreira and Henderson (1991).

Finally, **Pickering, Traxler, and Crocker (2000)** performed an eye-tracking study of sentences containing subordinate clause ambiguities. In this case, however, the verbs were held constant and the plausibility of the main-clause subject as an object of the verb was manipulated. They found faster right-bounded reading times in an ambiguous region immediately following plausible objects. There were also marginally significant effects on the noun itself. On the disambiguating verb, overall reading times were slower for plausible objects, and this effect emerged in the other reading time measures following the verb. These results indicate that subjects are sensitive to plausibility in this ambiguity. They are more likely to adopt an NP reading of plausible objects, resulting in faster initial reading and then a garden-path slowdown at the disambiguation.

3.5.1 Summary

Together, the studies reviewed here support the assertion that readers are sensitive both to verb argument biases and plausibility effects in resolving the subordinate clause ambiguity. As with the MV/RR and sentential complement ambiguities, it seems that we might expect an immediate reverse ambiguity effect in conditions that favor the early closure reading, followed by faster reading at disambiguation. Readers also have more difficulty comprehending early closure sentences when the length of the ambiguity is extended, although it is not clear how this should play out in terms of reading times at disambiguation. As argued by Ferreira and Henderson (1991), this finding may actually be an effect of increased distance between the head of the ambiguous phrase and the point of disambiguation, rather than of its overall length, but two failures to replicate this result using online measures justify a degree of skepticism.

3.6 Prepositional phrase attachment

Prepositional phrases (PPs) are a common source of ambiguity in English. The main ambiguities associated with prepositional phrases are determining the word modified by the PP and the semantic role that the PP plays. Although some work has been done on the ambiguous attachment of a PP to multiple NP sites (Cuetos & Mitchell, 1988; Gibson & Pearlmutter, 1994; Gibson, Pearlmutter, Canseco-Gonzalez, & Hickok, 1996), this review will focus on the ambiguity that has received the most attention in the literature: ambiguous VP versus NP attachment. This ambiguity is illustrated by sentences (32a) and (32b). The preposition *with* has been used in several experiments because it can fill a variety of roles in modifying either a noun (accompaniment, possession, physical feature) or a verb (instrument, manner). In (32a), *with binoculars* is generally interpreted as an instrument of *saw*, while, in (32b), *with a revolver* is generally interpreted as a possession of the *cop*.

(32a) The spy saw the cop with binoculars but the cop couldn't see him.

(32b) The spy saw the cop with a revolver but the cop couldn't see him.

The VP/NP attachment ambiguity has generated so much interest largely because it is thought to be a good test of the theory of minimal attachment. According to one model of phrase structure, but not all reasonable models (see Schütze & Gibson, 1999, for a discussion), the VP attachment uses fewer nodes. By the principle of minimal attachment, this should be the initially preferred reading regardless of semantic or lexical constraints.

Rayner, Carlson, and Frazier (1983) tested eye-movement reading time of sentences with a PP that modified either the main VP or the object NP, as in (32a) and (32b). They found slower first-pass and total reading rates for the NP-attached sentences in the region following *with*. Although this was attributed to a preference for minimal attachment, Schütze and Gibson (1999), as we will discuss below, have argued that the Rayner et al. (1983) results are better interpreted as a preference for *argument* over *modifier* attachments.

Taraban and McClelland (1988) questioned the Rayner et al. (1983) materials, suggesting that the sentences may have been biased toward the VP attachment. They created a new set of stimuli that were intended to be biased in favor of the NP attachment. The sentences were balanced in terms of word frequencies. Sentence completion and "goodness" rating tests confirmed that the Rayner et al. (1983) sentences were biased toward VP attachment while the Taraban and McClelland (1988) sentences were biased toward NP attachment.

Taraban and McClelland tested both their own sentences and those of Rayner et al.. As expected, the patterns of reading times were significantly different between the two sets of sentences. Both sets of VP-attached sentences were read equally fast and the slower reading times for the Rayner et al. (1983) NP-attached sentences were replicated. However, the new NP-attached sentences were read faster than the VP-attached ones, the opposite of the earlier result. The authors attributed this to differences in the degree to which subjects expected the noun in the relative clause. This is supported by the finding that expectation ratings were a good predictor of reading times. In a second experiment, Taraban and McClelland (1988) demonstrated that violations of subjects' expectations for the noun that is the object of the preposition had a small effect on reading times while violations in expectations of the role or attachment of the PP had greater and longer-lasting effects.

Sedivy and Spivey-Knowlton (1994) (see also Spivey-Knowlton & Sedivy, 1995) studied sentences with either a verb or noun-attached *with*-PP in which the PP followed an NP that contained either a definite or an indefinite article. It is expected that the indefinite NP should permit further modification and thus prefer the NP-attached reading. This expectation is supported by analysis of the Brown corpus, which found that PPs following definite NPs are more likely to be VP-attached and those following indefinite NPs are more likely to be NP-attached. Using phrase-by-phrase self-paced reading, the authors found longer reading times for NP-attached PPs following definite NPs than in the other three conditions. The interaction, however, was only marginally significant.

A sentence completion study of the same sentences, up to the PP, used in the first experiment revealed that the materials, which contained exclusively action verbs, were very biased toward VP attachment—more so than corpus analysis would support. A second set of materials was constructed using verbs of mental action or perception which, based on completions, had an overall NP attachment bias. In this case, VP-attached *with*-PPs conveyed manner, rather than instrument. With these materials, subjects still read the PP of sentences containing definite NPs faster when the PP was VP-modifying. However, for indefinite NPs, VP-modifying PPs were read *slower* than NP-modifying ones. Thus, the overall preference for VP attachment was eliminated and there was an NP attachment preference following indefinite NPs. These results indicate that, in both reading and sentence completion, subjects are sensitive to both verb type and NP definiteness.

According to one theory, PPs can act either as verb or noun *arguments* or as verb or noun *modifiers*, also known as adjuncts. Arguments fill a role that is implied by the modified phrase and take on a meaning specific to it, as in "John is a student *of physics*" or "Kim broke the window *with a ball*." Modifiers are not implied and do not change their meaning based on the modified phrase, as in "John is a student *from Phoenix*" or "Kim broke the window *on Sunday*." Schütze (1995) has collected a number of other properties or tests that can be used to help distinguish arguments from modifiers. Among others, these include the characteristics that arguments can normally modify a much narrower range of nouns or verbs, that modifiers can often be chained and can be more easily paraphrased, and that arguments generally must precede modifiers. Nevertheless, there remains some gray area in which the status of certain PPs as arguments or modifiers is not clear. Following Abney (1989), **Clifton, Speer, and Abney (1991)**, suspected that PP attachment preference may depend on whether the PP serves as an argument or a modifier of the phrase to which it is attached.

In a self-paced moving-window study that crossed attachment site with argumenthood, Clifton et al. (1991) found faster reading times for VP attachments but no effect of argumenthood on the prepositional phrase. However, on the subsequent phrase, which was the same across conditions, there was no effect of attachment site but faster reading for arguments over modifiers. An eye-movement study of the same sentences confirmed the faster reading times for VP attachments on the prepositional phrase.

In the subsequent region, all conditions were read at roughly the same rate except for the NP-attached modifiers. Although these data would seem to support an early preference for VP attachment, there are some reasons to question the findings. The authors were not careful to control for plausibility or expectations, as in Taraban and McClelland (1988). Konieczny (1996) (see Schütze & Gibson, 1999) argued that an analysis of the full data, not provided in the paper, suggests the initial preference was for arguments over modifiers. Schütze and Gibson (1999) pointed out several other potential problems, including phrases that differed in length and the use of common idioms.

Schütze and Gibson (1999) argued that PP attachment preference is mainly determined by whether the PP acts as an argument or as an modifier attachment and provided new data to support this hypothesis. Schütze and Gibson pointed out that, in the Rayner et al. (1983) materials, 8 of the 12 sentence pairs compared a VP argument attachment with an NP modifier attachment. This data might therefore be interpreted as reflecting an argument preference rather than a VP preference. The Taraban and McClelland (1988) materials, which found different results, actually included more cases in which VP-modifiers were contrasted with NP arguments. Argumenthood was not controlled for in a number of other studies, including Ferriera and Clifton (1986), Britt, Perfetti, Garrod, and Rayner (1992), Rayner, Garrod, and Perfetti (1992), Sedivy and Spivey-Knowlton (1994), and Britt (1994). If one regards PPs indicating instrument as arguments and PPs of manner as modifiers, the effects of action versus non-action verbs found by Sedivy and Spivey-Knowlton (1994) could also be attributed to an effect of argumenthood.

In their first experiment, Schütze and Gibson (1999) compared self-paced moving-window reading times on sentences with an NP-argument PP with those having a VP-modifier PP. Several different prepositions were used and stimuli were balanced for plausibility. They found no differences in reading time on the PP but much longer times on the phrase following it for VP-modifier PPs. The authors pointed out several possible confounds in this experiment, such as inherent complexity differences between NP arguments and VP modifiers.

They attempted to rule out these alternatives in a second experiment which included another condition with an unambiguous VP-modifier headed by a preposition that could not introduce a noun modifier and thus was meant to measure the inherent complexity of a VP-modifier. The unambiguous VP-modifier resulted in similar reading time to the NP argument, both of which were faster than the ambiguous VP-modifier. Thus, the conclusion was drawn that the slowness of this latter condition is not due to the inherent complexity of a VP-modifier or the inherent expectation of a noun modifier but to interference from the competing NP attachment interpretation.

Schütze and Gibson argued that the dominant preference in PP attachment is for arguments over modifiers, rather than for VP attachment over NP attachment. However, because they did not investigate NP-modifiers or VP-arguments, they cannot rule out the possibility that the VP/NP distinction plays an important role as well. The authors leave open the question, raised by MacDonald et al. (1994b), of whether argument preference may be due to argument attachments being more frequent in the language, rather than to a syntactically or semantically motivated preference.

3.6.1 Summary

Investigations of prepositional phrase attachment preference have found a variety of apparently conflicting results. Some studies claim to have found evidence of a VP attachment preference (Rayner et al., 1983; Clifton et al., 1991), while others have developed materials with no bias or a preference for NP attachment (Taraban & McClelland, 1988; Sedivy & Spivey-Knowlton, 1994; Schütze & Gibson, 1999).

A major problem seems to be that experimental materials have not been carefully controlled and effects may frequently be attributed to the wrong factors. A variety of factors are potentially relevant in PP attachment, including the VP versus NP attachment site, arguments versus modifiers, whether the verb takes an obligatory argument, and whether the NP permits further modification due to definiteness or the presence of other modifiers, not to mention a variety of other plausibility factors that bias subjects' expectations.

Because it seems clear that early VP attachment preferences can be eliminated or reversed through the manipulation of other factors, one could justifiably rule out a first-pass syntax-only parsing phase guided by minimal attachment. In fact, there seems to be no good evidence in support of an overall preference for VP attachment. A number of questions have been raised about the Rayner et al. (1983) and Clifton et al. (1991) studies that provided the best support for minimal attachment. Any bias in favor of VP attachment may well be due to the fact that VPs are more likely than NPs to be modified by a PP (see Section 4.6). Such general, statistically based, biases favoring VP attachment, if they exist, are probably weaker than other constraints.

Schütze and Gibson (1999) make a strong case for a dominant preference for arguments over modifiers in determining PP attachment. The results of Rayner et al. (1983), Taraban and McClelland (1988), and Sedivy and Spivey-Knowlton (1994) could all be attributed to a preference for arguments rather than for VP attachment or differences between verbs of action and verbs of perception. Schütze and Gibson (1999) also provided direct evidence supporting an effect of argumenthood. However, they could not rule out an additional effect of attachment site. It seems that the question of whether there exists a VP attachment preference independent of other effects remains open.

One problem with the theory that argumenthood is an important factor in guiding PP attachment is that the distinction between arguments and modifiers is not always clear. Although there are many clear cases, even proponents of argumenthood effects admit that there are gray areas. These gray areas may ultimately be filled in by measuring how subjects treat these PPs in reading experiments. In this case, the theory of argumenthood would change from an explanatory one to a descriptive one. It is likely that argumenthood per se is not what is guiding readers, but it happens to be highly correlated with a range of other factors that are actually guiding parsing, including contextual frequencies.

Finally, Sedivy and Spivey-Knowlton (1994) have shown that subjects are sensitive to the degree to which the NP permits further modification as determined by its definiteness. Subjects are slower to attach PPs to definite NPs. Britt (1994) also found that preceding the NP with an adjective causes a preference for VP attachment. Section 3.7 discusses these effects and a number of other studies of discourse influences on PP attachment.

3.6.2 Suggested experiments

- It would be informative to redo the Schütze and Gibson (1999) experiment with full crossing of the VP and NP attachments with the argument/modifier distinction. This would be the first study to compare VP and NP attachments while controlling for argumenthood. It would also be useful to include conditions with verbs that take obligatory third arguments (*I put the cup <u>on the table</u>*) and to replicate effects of NP modifiability in this better controlled environment.

3.7 Effects of discourse context

A number of studies have demonstrated influences of discourse context, that is, context preceding an ambiguous sentence, on comprehension of that sentence. These results support an expansion of constraint-based theories of sentence processing (MacDonald et al., 1994a; Trueswell & Tanenhaus, 1994) to include a broader range of constraints than those attributable directly to local lexical items. Experiments on discourse context are reviewed here because they are interesting. However, because the proposed CSCP model is limited to isolated sentences, it will be unable to address discourse-dependent phenomena for the time being.

3.7.1 Reduced relatives and sentential complements

Crain and Steedman (1985) found evidence of discourse influence in interpreting sentences involving structures temporarily ambiguous between relative clauses and sentential complements, as in:

(33a) The psychologist told the woman that he was having trouble with to leave her husband.

(33b) The psychologist told the woman that he was having trouble with her husband.

Sentences were placed either in a context that introduced just the one couple or in a context in which two couples had been introduced. *Referential theory* argues that the preferred analysis is not necessarily the syntactically simpler one but the one that is best supported referentially. Modifying a definite noun phrase presupposes the existence of more than one NP. In a context in which more than one NP of similar type have been introduced, a modifier must be used if one of those NPs is to be singled out later. Thus, the context with just one couple is expected to support the SC reading and the context with two couples is expected to support the RC reading.

Sentences were presented word-by-word at a fixed rate and subjects judged the grammaticality of the target sentences. 22% of the SC sentences were judged ungrammatical in a supporting context as were 54% in a non-supporting context. However, the RC sentences were judged ungrammatical about 50% of the time in either context. This has been taken as evidence that discourse plays some role in offline sentence parsing and that garden-pathing is not purely a function of the difficulty of a sentence but of its context as well.

Why the RC sentences were unaffected by context is not clear. Because the Crain and Steedman (1985) study did not use an online measure of processing complexity, it cannot distinguish rapid from delayed effects of context. The authors did stress the important point that a "null context" is not necessarily unbiasing. By not introducing any referents, a null context may bias the interpretation against readings that presuppose several referents.

Ferriera and Clifton (1986) used an eye tracker to test reading times on RR and MV sentences in either neutral or supporting contexts. MV and RR sentences differed only in that MV sentences had an additional *and* before the second main verb, as in (34b). There were no significant effects of context. RR sentences were read more slowly in the final region (*the story was big*) and marginally so on *agreed* vs. *and agreed*. This was taken as evidence against the role of context in initial parsing and in favor of minimal attachment. These results were confirmed in a phrase-by-phrase self-paced reading experiment using the same materials.

(34a) The editor played the tape agreed the story was big.

(34b) The editor played the tape and agreed the story was big.

Trueswell and Tanenhaus (1991) also studied the effects of discourse context on the MV/RR. Because it is set in the past tense, a MV sentence would not be appropriate in a future context. However, a reduced relative using a past participle can be used in a future context. Trueswell and Tanenhaus (1991) found that subjects are more likely to complete sentence fragments as MVs in a past tense context but as RRs in a future context. They also found that a future context leads to faster reading of RR sentences. In fact, reading time of reduced relatives in future contexts was similar to that of marked relatives. Trueswell and Tanenhaus (1991) also confirmed the "*only*-effect" found by Ni and Crain (1990). These results show that readers are able to make rapid use of discourse constraints in resolving syntactic ambiguities.

Trueswell and Tanenhaus (1992) replicated these findings and also added an unambiguous baseline condition. They found that, in past contexts, first-pass reading times for RR sentences were significantly higher than for marked sentences. But in future contexts, first-pass reading times for RR sentences were only slightly higher than for the marked controls. This demonstrates that a biased temporal context can nearly eliminate the difficulty of comprehending a reduced relative.

Mitchell, Corley, and Garnham (1992) compared self-paced reading times on sentences like (35a)–(35c) following contexts that supported either the RC or SC reading. In the second region, *had been*, sentences like (35c) were read more slowly than the others and this was not affected by context. In the last region, context and sentence type had only a marginally significant interaction. Spivey-Knowlton and Tanenhaus (1994) suggested that this failure to find effects of context may be due to the fact that the SC reading is overwhelmingly more frequent with the verb *told*. Judging by the examples provided in the paper, the SC-biased contexts in this experiment were rather weak.

(35a) The politician told the woman that he | had been | meeting | that he was going out to see the minister.

(35b) The politician told the woman that he | had been | meeting | the minister responsible once weekly.

(35c) The politician told the woman that | had been | meeting him | that he was going out to see the minister.

Consider the contexts in (35d) and (35e). Both contexts introduce two people, and mention that one of them had met with the politician before. Even after the SC-biased context, (35e), which introduces a man and a woman, it seems perfectly natural to find an RC sentence like (35a) which repeats the fact that the politician had been meeting the woman. The reader may have lost track of whether it was the woman or the man that had been meeting the politician and it seems normal to provide that clarification. A context that would provide better support for the SC interpretation would be one that introduced only a single person. Thus, the failure of Mitchell et al. (1992) to find early effects of context might be attributed to a weak context manipulation.

(35d) A politician was in his office waiting for an appointment. Eventually two women who were concerned about the environment turned up. The politician and one of the women had been meeting regularly but the other had only written to him before.

(35e) A politician was in his office waiting for an appointment. Eventually a man and a woman who were concerned about the environment turned up. The politician and the woman had been meeting regularly but the man had only written to him before.

Rayner, Garrod, and Perfetti (1992) compared eye-tracked reading time of RR and MV sentences in isolation and in favorable contexts. First-pass reading times at the point of disambiguation were longer for RRs than for MV sentences both in isolation and in context. Context improved first-pass reading times for both sentence types roughly equally. However, context had a greater effect on total reading time of RRs than it did for MVs. The authors argue that context had little or no influence on initial parsing decisions, but did affect reanalysis. The authors admit that the contexts used in this study did not contain the two possible referents for the head of the noun phrase that Altmann and Steedman (1988) argued were important for supporting the non-minimally attached interpretation. The sentences also contained some noun-verb pairs that are very much biased against the reduced relative, such as *the boy stood*, *the boy sat*, or *the legionnaires marched*. However, a more fundamental problem with this experiment is that reading times were being compared on disambiguating phrases that differed substantially across conditions. Slower reading for RRs cannot be attributed to garden-path effects.

Spivey-Knowlton, Trueswell, and Tanenhaus (1993) tested self-paced reading times of reduced and marked relatives in contexts biased toward either the MV or RR interpretation. Using a two-word moving window, Ss read reduced relatives more slowly than marked ones at the *verb+by* region in MV contexts but not more slowly in RR contexts, although the interaction was not significant. In the following region, however, the interaction was reliable. In a second experiment, the authors compared ambiguous and unambiguous RRs in similar contexts as before. In this case, the interaction with context was significant at the *verb+by* region, supporting an immediate effect of referential context. In word-by-word self-paced reading, in which *by* was not parafoveally available upon reading the verb, the interaction did not reach significance.

Spivey-Knowlton and Tanenhaus (1994) point out that referential contexts need to be quite strong to have an effect on the MV/RR ambiguity. In particular, contexts cannot simply introduce two NPs in order to support the reduced relative interpretation. They must mention that something, the action in the reduced relative, happened to only one of the actors. The Ferriera and Clifton (1986) study may have failed to find effects of context because its contexts, in introducing the action, used verbs that differed from the ambiguous verb in the test sentence. Furthermore, as we have mentioned, contexts supporting the MV interpretation are best if they do not introduce two NPs, even if those NPs are distinguished in more than one way.

3.7.2 Prepositional phrase attachment

Ferriera and Clifton (1986) also included a PP attachment condition in their study of context and found no effects of context. In first-pass reading times, NP-attached sentences were read more slowly on the phrase following the ambiguous PP. The minimal attachment effect was greater in second-pass reading times. However, Britt (1994) points out that five of the eight verbs, such as *put*, took obligatory goal arguments. Thus subjects were expecting a VP-attached PP so that a garden-path effect on the intervening NP-attached PP is hardly surprising. Britt (1994) also noticed that the contexts used in Ferriera and Clifton (1986) tended to bring one of the NPs to the foreground prior to the ambiguous sentence. As a result, further specification of the NP was not necessary and the NP-attached PP was not expected.

Altmann and Steedman (1988) found evidence against minimal attachment in an experiment involving attachment of prepositional phrases headed by *with*. Test sentences were ambiguous between NP and VP attachment of the PP, but were semantically biased toward one or the other. Sentences were also preceded by a context that optionally supported the NP attachment, by introducing more than one NP, or the VP attachment, by introducing more than one possible instrument for the action. These factors were treated independently. Reading time was measured for the sentence as a whole.

Overall, the "minimally-attached" VP-attached sentences had *longer* reading times than the NP-attached ones. In the most VP-supporting context, their reading times were about the same. In an NP-supporting context, the reading times of NP-attached sentences decreased and those of VP-attached sentences increased. Subjects appear to have been sensitive to the number of NP referents in the context, but not to the number of potential instruments. These results were confirmed in a second experiment that measured phrasal rather than whole-sentence reading times.

Rayner, Garrod, and Perfetti (1992) tested eye-tracked reading times of VP- and NP-attached PPs in isolation and in favorable contexts. Rather than introducing referential ambiguity, the contexts in this experiment varied whether or not the referent was focused. First-pass reading times at the disambiguating phrase were longer for NP attachments both in isolation and in context. Favorable context improved reading times of both sentence types roughly equally. If anything, there was a slightly greater improvement for NP attachments. This would seem to suggest that VP attachments are easier overall. However, whether favorable contexts can remove this bias is unresolved. Sentences should have been tested in neutral and unfavorable contexts as well. Isolation leads to longer reading times across the board, and is thus not a good neutral baseline. Also, the methodology in this experiment involved comparing reading times of different disambiguating phrases which, as we have said several times, is not a good idea unless that phrase difference is the variable of interest.

Britt, Perfetti, Garrod, and Rayner (1992), on the other hand, found somewhat different results in their reading-time study of ambiguous VP- and NP-attached PPs. The sentences themselves favored just one interpretation and followed either a neutral context or one supportive of the correct interpretation. Sentence presentation was self-paced and proceeded either phrase-by-phrase or with a one-word moving window. Both presentation modes achieved the result that, at the point of disambiguation, sentences with VP-attached PPs in either context and those with NP-attached PPs in good contexts were read equally fast. Only NP-attached PPs in neutral contexts were slower. Thus, subjects were able to make use of context in reading the less-preferred NP-attached PPs. Regrettably, the authors did not include an unfavorable context condition. Unlike Altmann and Steedman (1988), but in agreement with Rayner et al. (1992), this study suggests that VP-attached PPs are easier.

A second experiment used eye-tracking and a slightly different set of sentences. NP attachment was either indicated by a phrase after the PP, as in (36b), or by the PP itself, as in (36c). In the first experiment, only cases like (36b) and (36c) were used. Unbiased sentences were read in isolation, rather than with a neutral context. With a biased context, all sentences were read equally fast. In isolation, sentences like (36b) were read more slowly at the final phrase and sentences like (36c) were read more slowly at the PP. Overall, reading times were a bit higher for sentences in isolation. This experiment confirms the notion that VP-attached sentences are easier in neutral context but that this advantage is eliminated by supportive context.

(36a) Peter read the books | on the chair | instead of lying in bed.

(36b) Peter read the books | on the chair | instead of the other books.

(36c) Peter read the books | on the war | instead of the other books.

A third experiment also looked at reading times of VP- and NP-attached PPs in context and in isolation as well as reduced relative sentences. In this case, although NP-attached sentences were again harder in isolation, reading of both VP- and NP-attached sentences was aided by context. The disambiguation region was read more slowly after reduced relatives than after VP-attached PPs with or without context. The authors conclude that there was no context override for RRs as there was for PPs and that, "discourse information is not readily used in parsing reduced relatives." (p. 310)

However, this conclusion obscures the fact that RRs, as well as PPs, were read significantly faster in context. In fact, the difference in reading time between biased and unbiased contexts was numerically greater for RRs. Given that the disambiguating regions differed for the two sentence types, a direct comparison of reading times of these regions is inappropriate. It thus seems that Britt et al. (1992) have, if anything, found evidence that *both* PPs and RRs are aided by prior context. The extent to which VP-attached PPs are aided by context seems questionable. Spivey-Knowlton and Tanenhaus (1994) also found, in a sentence completion study of the Britt et al. (1992) materials, that subjects completed sentences in relative-clause biasing contexts as main verbs 83% of the time. Thus, their finding may be partially due to weak context.

Finally, **Britt (1994)** investigated some issues that were left open in the previous studies of discourse effects on the PP attachment ambiguity (Ferriera & Clifton, 1986). Britt used two verb-type conditions: one in which the verb required an obligatory goal and one in which the verb permitted an optional goal. There were also three context conditions: a one-referent condition, a two-referent condition, and a two-referent condition that primed the fact that only one of the referents would be chosen in the ambiguous sentence. Once again, sentences either contained VP-attached or NP-attached PPs.

In a self-paced, moving-window task, subjects were slower at reading the disambiguating material (the NP in the PP and the following word) for NP attachments in all cases except in both two-referent conditions with optional-goal verbs. Thus, subjects preferred VP attachment whenever the VP required a goal and when the discourse only introduced one referent. There was no difference between the two-referent and two-referent primed conditions. In another experiment, Britt (1994) showed that preceding the NP with an adjective that distinguishes it causes a VP attachment preference, even in a two-referent context. A reasonable conclusion is that once a clear reference is established, subjects do not expect further modification.

3.7.3 Summary

Quite a few studies have failed to find early effects of discourse context in resolving the MV/RR ambiguity or the SC/RR ambiguity (Crain & Steedman, 1985; Ferriera & Clifton, 1986; Mitchell et al., 1992; Rayner et al., 1992). However, the strength of the context manipulation used in these studies has been criticized. Four other papers (Trueswell & Tanenhaus, 1991, 1992; Britt et al., 1992; Spivey-Knowlton et al., 1993) did find early effects of discourse context on the MV/RR ambiguity. Trueswell and Tanenhaus (1991, 1992) found that future contexts, which support the RR interpretation, eliminated or greatly lessened the reduction effect on reading time for relative clauses. Spivey-Knowlton et al. (1993) found significant immediate effects of contexts that manipulated the number of possible referents. Although readers are sensitive to context influences in resolving the MV/RR ambiguity, it is clear that the context manipulation must be very strong to, by itself, eliminate the reduction effect.

Less work has been done on the effects of context on sentential complements. Mitchell et al. (1992) found only late effects of context in ambiguous SC/RR sentences. However, their context manipulation appears to have been weak.

Crain and Steedman (1985) did find effects of context on grammaticality decisions, but it is impossible to say whether these were the result of early or late influences in the parsing process. While one might expect that discourse context should have an immediate effect on the NP/S ambiguity, there is as yet no data to support this.

There is better evidence in favor of discourse effects on the PP attachment ambiguity. Altmann and Steedman (1988), Rayner et al. (1992), and Britt (1994) found effects on both NP- and VP-attached PPs. Britt et al. (1992) found effects for only NP-attached PPs in one experiment and for both NPs and VPs in another. Only Ferriera and Clifton (1986) failed to find a significant context effect, but their choice of verbs and the strength of their context manipulation has been questioned (Britt, 1994).

Again, these studies do not paint a clear picture regarding an overall preference for NPs or VPs. Altmann and Steedman (1988) found an overall preference for NP attachment in their materials, but Britt et al. (1992) and Rayner et al. (1992) found the opposite. Again, it seems that an independent effect of attachment site, if it exists, is probably much smaller than other influences, such as discourse effects, argumenthood, verb type, and NP modifiability.

3.8 Production

All of the data discussed so far has been in the realm of comprehension. It is the goal of the CSCP model to accurately reflect human performance in production as well as comprehension. Unfortunately, there is relatively little data on adult production of relevance to the model. This is largely because of the difficulty of conducting sentence production experiments, and also because much of the production work that has been done has dealt with discourse level or prosodic phenomena which cannot be addressed with the model. However, two areas that have received some attention are spontaneous speech errors and the structural priming phenomenon.

3.8.1 Speech errors

Noticeable errors are quite rare in the everyday speech of unimpaired adults. Nevertheless, those that do occur provide some insight into our language production system. In order to obtain an accurate statistical estimate of the frequency of various types of speech errors, one would need to analyze a large volume of spontaneous speech data. Even so, many errors, semantic substitutions in particular, could go undetected, since we do not generally know what a speaker was intending to say. However, researchers seem to agree that most speech errors fit into a few common categories, and it may be sufficient to see if the CSCP model makes types of errors similar to those made by humans.

Bock (1990) identifies four main forms of speech errors: word exchanges, semantic substitutions, phonological substitutions, and morpheme exchanges. Morpheme exchanges and phonological substitutions both operate at the sub-word level. Although interesting, they are not as relevant for the CSCP model because it does not seek to address phenomena operating at a finer grain than the production of whole words.

Semantic substitutions involve replacing a word with one having similar meaning. Most examples of semantic substitution in the literature are noun replacements. However, earlier versions of the CSCP model also seemed to produce a lot of verb substitutions, as well as changes of tense without changing the verb. It will be interesting to see if semantic verb substitutions actually occur in human data. It may be that these are more often overlooked than noun substitutions.

Word exchanges involve the swapping of two words within the sentence. **Bock and Loebell (1990)** note that the majority of word exchanges do not cross clause boundaries. This has been taken as evidence that sentence production operates on clause-sized units. Two other interesting trends govern word exchanges. Exchanges nearly always happen between words of the same form class. Nouns exchange with nouns, adjectives with adjectives. As a result, the words tend to occupy positions appropriate for their form class and production errors tend to retain valid syntactic structure.

In cases where word exchanges would cause violations in case or agreement, a phenomenon called syntactic accommodation often occurs. Verbs will change in number to maintain agreement with nouns, and case marked words, such as pronouns, might change case. For example, “He chases girls,” is more likely to come out as “Girls chase him” than as “Girls chase he.”

Although they are less common, speech errors do occur in which syntactic violations are made **(Stemberger,**

1982). These sometimes involve exchanging neighboring words, possibly using the phrase structure for a statement within a question, as in (37a). Another common form of syntactic error is what Stemberger called *blends*. These may be characterized by redundancy in cases where two different word orderings are possible, as in (37b) and (37c).

(37a) Uh-oh, where it is?

(37b) Turn on the light on.

(37c) This is getting very difficult to cut this.

While speech errors at the word level are interesting, they cannot provide a very rigorous test for a speech production model for several reasons. They are rare enough to make gathering large collections of errors difficult, certain types of errors are more likely to go unnoticed than others, biasing frequency counts, and syntactic errors are hard to classify and may result from combinations of several factors. So it is difficult to say whether a model is performing appropriately.

3.8.2 Structural priming

An issue that has come to the forefront of interest in speech production work in recent years is structural or syntactic priming. This is the phenomenon by which speakers tend to repeat syntactic forms that they have recently used or heard. Although it has long been noted that people tend to perseverate a particular pattern of speech, this effect was not produced under experimental conditions until fairly recently.

Bock (1986) elicited structural priming as follows. Subjects heard and then repeated a prime sentence. They were then shown a drawing depicting an event and were asked to give a one sentence description of that event. Of interest was whether the syntactic form of the priming sentence affected the form of the picture description. These trials were embedded in filler materials and subjects were unaware of the purpose of the experiment.

In the first study, two syntactic distinctions were examined: actives versus passives and prepositional versus double object datives. Prime sentences used one of the four forms. The results of this experiment were quite impressive. Prepositional dative primes resulted in significantly more prepositional dative than double-object utterances and double-object primes resulted in significantly more double-object utterances. Because of the overwhelming bias in favor of active utterances, the priming effects for voice were not as strong. However, there were more active responses to active primes than to passive primes and more passive responses to passive primes. Although all of the elicited prepositional datives used *to*, primes using *for* also resulted in the priming effect, albeit not as strongly as with *to*. This suggests that priming can occur across small variations in surface form.

Two additional experiments varied the animacy of the agent in the active and passive prime sentences and found no effect of prime animacy. This supports the conclusion that priming occurs across variations in meaning. These experiments did find significant priming effects on the production of passives but not on the production of actives, although there were trends in the right direction. They also found significant differences in whether actives or passives were used depending on the animacy of the agent in the pictured event. Human agents resulted in many more active descriptions.

However, **Bock, Loebell, and Morey (1989)** studied active/passive primes with animate and inanimate subjects and did find a significant effect of animacy. In this case, all of the pictured events had inanimate agents. Primes with inanimate subjects, both active and passive, resulted in a higher percentage of active productions than did primes in similar voice with animate subjects. This would appear to contradict the findings of Bock (1986), but the manipulation was not quite the same in this experiment. Bock (1986) varied whether the *agent* of the prime was animate, with an equal split between animate and inanimate patients. Bock et al. (1989), on the other hand, varied whether the *subject* of the primes was animate. The object was always inanimate when the subject was animate and vice-versa. Thus, the lack of an animacy effect on priming in Bock (1986) may have been due to the fact that patients in the prime sentences had mixed animacy. At any rate, it appears that animacy can play some role in structural priming. What is not clear is exactly which conditions should result in priming and which should not.

Bock and Loebell (1990) conducted three interesting structural priming experiments. The first compared double-object primes with prepositional datives and prepositional locatives, both using the preposition *to*. Both types of prepositions effectively primed prepositional dative constructions. This suggests that priming may be insensitive to conceptual differences. However, the difference between a dative and a locative is relatively small and arguably not a major conceptual one.

A second experiment was conducted in which prepositional locatives using *by* were compared with passives having an agent *by* phrase. Control primes were active sentences. Both passives and prepositional locatives primed passive descriptions more than did actives. There was no significant difference between the effects of passives and locatives. This helps confirm that conceptual differences do not interfere with structural priming. However, it is not clear from these two experiments whether priming was based on the surface forms or deep structures of the primes. In both cases, the effective primes shared the same surface form.

The third experiment attempted to rule out the importance of surface forms by using two priming conditions that shared surface form but differed in deep structure. Priming conditions consisted of prepositional datives, (38a), infinitives, (38b), and double-object datives, (38c). The prepositional datives and infinitives both used a *to* phrase, but they differed in deep structure.

(38a) The defendant told a lie to the crowded courtroom.

(38b) The defendant told a lie to protect his daughter.

(38c) The defendant told the suspicious lawyers a lie.

In this case, only the prepositional datives effectively primed other prepositional datives. This suggests that the important factor in the earlier experiments was similar deep structures, rather than similar surface forms. However, one could argue that the prepositional datives and infinitives did not really share common surface structures. In the one case, the *to* was followed by a noun phrase and in the other it was followed by a verb phrase.

All of the syntactic priming experiments discussed thus far had subjects both comprehend and reproduce the prime sentences. The resulting priming could arise from just one or both of these processes. **Branigan, Pickering, and Cleland (2000)** found evidence of priming resulting from comprehension alone. Subjects performed a task in which they described pictures to another person, who was actually a confederate of the experimenter, and selected pictures based on descriptions given by the confederate. The confederate controlled whether he or she used double object or prepositional datives and the form of the confederate's description was found to affect the form of the subject's subsequent description of a different picture. In this case, the magnitude of the priming was much greater when prime and target shared the same verb.

One final issue that has been examined is the persistence of structural priming effects. **Bock and Griffin (2000)** used similar materials as did Bock (1986) but varied the distance between prime and target by introducing from 0 to 10 filler items. The priming effect was undiminished even after 10 intermediate items. However, there was a mysterious lack of priming following 4 intermediates. Priming for transitives (active vs. passive) was weaker and shorter-lived than for datives. The persistence of structural priming suggests that it may be due to a learning mechanism rather than to activation-based memory (Dell et al., 1999; Chang, Dell, Bock, & Griffin, 2000).

3.8.3 Summary

Structural priming is a robust phenomenon that can be induced by comprehension as well as production and can last for a fairly long time. Priming is more effective when the frequency of alternative structures is balanced, as with the prepositional versus double object dative, than when one structure is generally preferred over the other, as with actives and passives. Structural priming occurs despite variations in meaning or the particular open-class words being used.

It seems that animacy, which is considered by some a semi-syntactic variable, can affect the priming of voicing. However, whether the animacy of subjects versus objects or agents versus patients is the most important factor seems, as yet, unresolved.

Also unresolved is the importance of shared surface structure. Most of the successful priming experiments use primes and targets with similar surface structures. The one experiment that attempted to rule out the importance of shared surface structure seems to have actually used rather different surface structures (Bock & Loebell, 1990, Experiment 3). A simple change of preposition does not eliminate priming, but it does reduce it (Bock, 1986). In written production, Pickering and Branigan (1998) found that the priming effect is reduced when different verbs are used in prime and target. It therefore seems likely that surface structure affects priming, although it is probably not solely responsible for it.

3.8.4 Suggested experiments

- The extent to which structural priming depends on surface forms does not seem to be completely resolved. The third experiment of Bock and Loebell (1990) found no priming between prepositional datives and infinitives. However, these arguably do not share surface forms. A better experiment might be to compare the priming of NP-attached, (39a) and (39c), and VP-attached, (39b) and (39d), PPs on prepositional datives.

  (39a) The dog chased the assistant to the principal.
  (39b) The dog chased the assistant to the outhouse.
  (39c) The spy saw the policeman with a revolver.
  (39d) The spy saw the policeman with a telescope.

  If constituent structure is the important factor, only (39a) and (39c) should prime sentences with NP-attached PPs, regardless of the preposition used. If surface structure is important, we should see priming for NP-attached PPs on VP-attached sentences which use similar prepositions.
- The studies by Bock (1986) and Bock et al. (1989) had largely contradictory findings regarding the animacy effects in active/passive priming. The latter study found significant effects while the former did not. To clarify these issues, one would wish to see an experiment that contained eight priming conditions, crossing voicing with the animacy of both agents and patients. Ideally, there should also be four conditions for the pictured events, crossing the animacy of the agents and patients.

3.9 Summary of empirical findings

This section briefly summarizes the empirical findings discussed in this chapter and describes how the corresponding experiments will be conducted with the CSCP model. Under **Effects** are summaries of the apparently reliable patterns of data that we hope the model to capture. References can be found in the Summary sections above. Because the experimental results reviewed here are, in general, so inconsistent across studies, it will not be worthwhile at this stage to attempt to match specific numerical results with the model. The best we can say for most of the results is that a certain variable should have no effect or an effect in a particular direction, and, in some cases, that certain effects should be stronger than others.

Variables lists the statistical variables that must be controlled in the language on which the model is trained to reasonably perform the intended experiments. The data for most of these variables will be taken from the Penn Treebank (Marcus et al., 1994) corpus of written text and will be used to construct a language for the model that is similar to English in important respects. The extraction of that data and the construction of the training language are the topics of the next two chapters. The **Not addressed** sections mention some interesting structures or empirical effects discussed above that will not be handled in the current version of the model.

3.9.1 Relative clauses

It will be interesting, first of all, to see how well the network is able to make use of deep structure to correctly comprehend sentences with embedded relative clauses. For example, if faced with, "The girl that kissed the boy blushed," (from Bock & Loebell, 1990), would the model interpret "the boy blushed" to mean that the boy, too, was blushing? How will the model respond to the queries, "Who blushed?", "What did the girl do?", and "What did the boy do?"

Effects

- Single object-relatives are more difficult than single subject-relatives. Passive relative clauses should also be tested. These are quite common in natural language, but have never before been tested, except in the context of the MV/RR ambiguity. Based on the model's performance, a prediction will be made about the expected difficulty of passives relative to the other RC types.

- Single right-branching RCs are a bit more difficult than center-embedded RCs. This effect should be considerably weaker than the clause-type effect.
- If one relative-clause is embedded within another, the added difficulty should be a function of whether the innermost clause is subject- or object-relative.
- Reduction should make nested $\mathtt{CO}'^2$ sentences more difficult than $\mathtt{CO}^2$s, and should possibly improve COSs as well. Reduction effects for other relative clause types are not well established and any significant differences will be an important prediction of the model.
- Although there is not much direct evidence one way or the other, we expect semantic constraints, animacy in particular, to have a beneficial effect on the comprehension of single and double relative clauses. The difference in the model's performance on sentences with no semantic constraints (in which every noun is an equally good agent and/or patient for every verb) and those with strong constraints will be another testable prediction.
- Relative clauses with indexical pronouns should be easier than those with full NPs or other pronouns. It is not clear if this should be attributable to referential effects or simply to frequency effects.

Variables

- The frequency (F) of modification of sentence subjects and sentence objects by subject-relative, object-relative, and passive RCs.
- The Fs of embedding the three RC types within another RC.
- The Fs of reduction for the three RC types, in isolation and nested within other RCs.
- In order to model the indexical pronoun effect, the training language will include them. Ideally, it would also include pronouns that make reference to NPs in other clauses of the same sentence. The Fs of these should be based on conversational corpora, if available.

Not addressed

Multiply-embedded relative clauses, beyond one relative clause nested within another, will not be included in the language. These are exceedingly rare in natural language and we can't say too much about them except that the general public has serious difficulty with them and may not even consider them grammatical.

Gibson and Thomas (1995) have some interesting data regarding acceptability differences between noun-modifying sentential complements nested within relative clauses and their reverse. Because these are so rare (a sentential complement within a relative clause almost never occurs) in natural language, they will not be included in the training language. It may be reasonable to address this phenomenon in a later version of the model.

3.9.2 The MV/RR ambiguity

The ability of the network to comprehend any sort of ambiguous reduced relative will be interesting in and of itself. It will be particularly informative to analyze the intermediate parsing representations used by the network during and after the relative clause, comparing representations formed on ambiguous and unambiguous sentences.

However, it is also important that the model demonstrate sensitivity to the factors that are known to be relevant to this ambiguity. Unfortunately, it will not be possible to include enough verbs in the model's language to independently account for the effects of verb frequency as a past participle, with a transitive argument structure, in the passive voice, and with reduced (as opposed to marked) RCs. Therefore, this first experiment will simply compare verbs that frequently occur with reduced relatives to those that do not. Although the more specific factors will contribute to a verb's RR frequency, their effects will not be analyzed in isolation. To do so would require enough verbs to independently vary each factor while controlling for all others, which is beyond the scope of the current investigation.

In modeling the MV/RR data, as well as the results from other ambiguities, it will be necessary to extract a surrogate for reading time from the network's internal states. This should involve measures derived from the network's predictions as well as measures of the complexity of state transitions in its hidden layers. Larger state changes may reflect processing difficulties that, in humans, would manifest themselves as reading delays.

Effects

- In general, reduced relatives should be read more slowly and produce more comprehension errors than marked relatives and MV sentences.
- The disadvantage of ambiguous RRs, in comparison to either RRs with unambiguous verbs or marked relatives, should decrease or disappear with verbs that frequently occur with reduced relatives.
- Semantic constraints, either depending on animacy or when controlling for animacy, should also ameliorate the reduction effect.
- On sentences heavily biased toward the RR interpretation, we may expect to find a temporary reverse ambiguity effect: faster reading at the point of disambiguation but slower reading on or immediately following the verb.

Variables

In order to conduct these experiments we will need to include some verbs that have distinct past tense and past participle forms, as well as a number that do not.

- The F with which the selected verbs occur with transitive argument structures.
- The F with which the verbs occur in passive constructions.
- The F with which passive relative clauses using each verb are reduced.
- The F with which each noun occurs as the subject and object of each verb. This will not necessarily be based on corpus data, but reasonable approximations will be made so that experiments on semantic effects can be performed.

Not addressed

As mentioned before, we will not attempt to account for the individual contributions of specific verb-dependent factors, such as transitivity and passivity. Not only would those variables be difficult to control, there does not seem to be any good data on the independent contributions of the variables in humans. Thus, results could not be immediately evaluated. It may be useful to examine these factors in a large-scale experiment conducted on humans and, using the same verbs, on the model.

Although reading span has been shown to have a significant effect in resolving this ambiguity, that too will not be accounted for with the model. However, it may be possible to match networks with different levels of training or resources to high- and low-span readers.

3.9.3 Sentential complements

In order to perform the SC experiments, a number of verbs of thinking and communicating will be included in the language. Most will permit either NP or sentential complements but some may allow only SCs.

Effects

- The sentential complement reduction effect should be stronger for NP-biased verbs. SC-biased verbs should show a minimal effect of reduction and possibly a reverse reduction effect.
- The verb-bias effect should be stronger for reduced SCs.
- Noun semantics should play a role in this ambiguity, but should have a lesser effect than verb bias. It is not clear whether the effect should be stronger for SC- or NP-biased verbs.
- It would be interesting if an early reverse ambiguity occurs under strongly SC-biased conditions, but the existence of this is not fully confirmed in humans.

Variables

- The F with which each verb is followed by an NP or sentential complement.
- The F with which the SC is reduced following each verb.

3.9.4 Subordinate clauses

In addition to the basic, unpunctuated subordinate clause, (40a), it will be interesting if the model is able to comprehend more complex cases, like (40b) in which the main clause contains a relative clause.

(40a) Before the dog attacked the boy ran away.

(40b) Before the dog attacked the boy the dog chased ran away.

Effects

- In the basic unpunctuated subordinate clause ambiguity, the model should show sensitivity to verb transitivity and the plausibility of the noun as the object of the verb.

Variables

- The F of verbs taking intransitive and transitive argument structures.
- The F of various types of subordinate clause, even though this may not be relevant to the effects.

3.9.5 Prepositional phrase attachment

It will be interesting to see if the CSCP model is even able to correctly resolve PP attachment ambiguities. If it is generally successful, we can then examine which factors affect its success and its reading time measures.

Effects

- There should be a preference for argument over modifier attachment for both VPs and NPs.
- VPs, like *put*, that take an obligatory third argument should have preferred attachments.
- Attachment to an NP should be harder if the NP is definite or already has adjective modification.
- When controlling for argumenthood, whether VPs or NPs have preferred attachments overall will be a prediction of the model.

Variables

- The overall F of VP- and NP-attached PPs.
- The specific Fs of different PP types for each verb.
- The overall F of arguments and modifiers of nouns.
- The probability that a noun is modified by a PP given its definiteness and whether it is also modified by an adjective.

Not addressed

There are some interesting results associated with PPs that are ambiguously attached to multiple NP sites (Cuetos & Mitchell, 1988; Gibson & Pearlmutter, 1994; Gibson et al., 1996). However, such cases will not be included in the current study because of the difficulty of incorporating them into the training language in a reasonable way. Situations in which an ambiguous NP-attachment arises are relatively rare and the results probably depend on semantic and statistical constraints that would be hard to control and would require a great number of nouns.

Consider examples (41a) and (41b) from Gibson and Pearlmutter (1994). A headmaster can only be *of* a school or something very much like one, and a school can only be *for* a few things, such as boys, girls, dogs, or the gifted. Similarly, a building is one of the few types of models that a computer could be near. Thus, the relations in these sentences are highly constrained semantically and the nouns are not very productive. In order to respect the semantic constraints of English that may crucially underly NP attachment effects, one must prevent the training language from containing nonsense like, "The school of the computers near the headmaster." To do so will require a very large number of nouns and careful attention to reasonable NP-preposition-NP combinations.

(41a) The headmaster of a private school for boys...

(41b) The computer near the models of the buildings...

3.9.6 Production errors

- Production errors should consist mostly of word exchanges and semantic substitutions.
- Word exchanges should mostly occur within a single clause.
- Word exchanges should mostly occur between words of the same form class.
- Syntactic accommodation should occur with word exchanges.
- Production errors should rarely violate grammaticality. Repetitions in cases with alternative phrasings are one situation in which they might.

3.9.7 Structural priming

- Structural priming should be inducible through either comprehension or production.
- Priming should occur for both datives and transitives, but be weaker for transitives because of the general preference for actives over passives.
- Priming should occur across different prepositions, from prepositional locatives to datives, and from locatives to passives.
- Priming should occur across verbs, but be diminished when the verbs differ.
- The animacy of subjects in transitive primes should have an effect, as in Bock et al. (1989).
- Priming should last across several intervening sentences.

3.9.8 Other effects

- The model should be able to resolve unambiguous sentences that contain lexical ambiguities. It will be interesting to see how it responds to globally ambiguous sentences like, "They can fish."
- It will be interesting to see how well the model can paraphrase, that is, comprehend a sentence and then reproduce it based only on the message it derived.
- The model should be able to handle several different forms of the word *that*, which can act as a pronoun, a demonstrative pronoun, a sentential complement determiner, and a relative pronoun.

- Lewis (2000a) gives an example of a sentence fragment, “Mary suspected the students who saw her yesterday. . . ,” that is ambiguous in three ways and has eight different syntactic completions, none of which seem too difficult. It may be possible to test the ability of the model to handle this ambiguity if we allow the training language to contain possessive pronouns and an SC within an RC within an SC. Even if we limit the depth of embedding to two, we could at least test the model on the four-way ambiguity.

Chapter 4

Analysis of Syntax Statistics in Parsed Corpora

The previous chapter reviewed a collection of empirical studies of sentence processing spanning topics that involve a number of syntactic structures which are of particular interest to researchers because they give rise to ambiguity or other difficulties. A variety of factors were found to affect how people process these structures, but one set of factors that is consistently thought to be relevant is the frequency with which various structures occur in language or the frequency with which particular words or combinations of words occur in conjunction with those structures. For example, the difficulty of the main verb/reduced relative ambiguity is thought to be influenced by the overall frequency of reduced relatives versus main verb constructions and the frequency of the verb in passive voice, in a relative clause, and in a reduced relative.

A model that is sensitive to frequency effects must be trained on a language representative of the target language, English in this case, in terms of any frequencies that may be relevant to the phenomena of interest. Ideally one would simply expose the model to a linguistic environment that is similar to that experienced by an average English speaker over the course of his or her lifetime. This is currently infeasible for any number of reasons, not least of which being that we don't really know much about the true distribution of sentence types to which people are exposed over the course of their lifetimes. Therefore, rather than using full-fledged English, the model must be trained on an artificial language that only approximates English. But in order for the model to replicate human performance in an appropriate way, the frequencies of relevant constructions in the artificial language must match those in English.

This chapter, therefore, consists of an analysis of various syntax statistics in English. In Chapter 5, some of the results of this study will be used in constructing the artificial language on which the CSCP model will be trained and tested. While analyzing purely lexical statistics in large corpora is now relatively easy, gathering statistics over syntactic structures remains a difficult task. The Penn Treebank project has solved much of the problem by painstakingly tagging several large English corpora with syntactic and part-of-speech markings (Marcus et al., 1993, 1994). But due to the variability of natural language, tagging errors, and the inevitable inadequacy of any tagging system, extracting data from the tagged corpora remains a complex and time-consuming process. This chapter presents an analysis of the frequency with which various types of syntactic structures occur in the Penn Treebank's Wall Street Journal and Brown corpora (Kucera & Francis, 1967) of English text, with particular emphasis on verb phrases and relative clauses.

Section 4.1 introduces some problems encountered when extracting statistics from parsed corpora and the methods used here to overcome these problems. Section 4.2 analyzes the verb phrases in the Wall Street Journal and Brown corpora including a specific analysis of tense and voice frequencies. Section 4.3 deals with relative clauses including effects of extraction type (subject extraction, object extraction, or passive), whether the relative pronoun is present, and whether the clause modifies a subject or object noun. Sections 4.4–4.7 cover sentential noun phrases, prepositional phrases, and coordination and subordination, respectively.

4.1 Extracting syntax statistics isn't easy

The standard tool for extracting information from the Penn Treebank is the TGREP program, which selects from a corpus phrases that match a specified structural pattern. While this works well for simple constructions, designing patterns for more complex ones can be quite difficult. A first attempt at constructing a pattern or set of patterns for isolating a structure tend to result in either over- or under-generalization. The pattern either fails to match some desired structures, matches some undesired ones, or both. This is largely due to the natural and erroneous variability in the encoding of structures in the Treebank.

Natural variations include possible alternative choices in phrase ordering or the insertion of optional structures, such as prepositional phrases, that might disrupt the typical pattern. A particularly common and insidious problem is exemplified by the effect of composite phrases. For example, using the Treebank II tagging style (Santorini, 1990), a simple past tense passive would ordinarily be encoded with a structure like the following:

- `(VP (VBD was) (VP (VBN fired)))`

Such a structure might be matched by a pattern that looks for a `VBD` containing *was* which is followed by a `VP` containing a `VBN`. However, a conjunction creates an additional level of `VP` between the `VBD` and the `VBN`:

- `(VP (VBD was) (VP (VP (VBN fired)) (CC and) (VP (VBN prosecuted))))`

In this case, neither fired nor prosecuted will match the pattern, although an accurate accounting, arguably, should include both. Because TGREP does not provide a convenient method for encoding a chain of nested symbols, such variations typically must be handled with special cases or additional patterns.

Furthermore, structures in the Penn Treebank are frequently either incorrectly marked or marked inconsistently due to judgment calls on the part of the taggers or variations in neighboring syntactic structures. For example, past participles, such as *abducted*, are variously marked as verbs in the simple past tense, as adjectives, or as modifying nouns. In the Brown corpus, which is tagged in the Treebank I style, auxiliary verbs are generally marked using a separate `AUX` structure preceding the main `VP`, but are also occasionally found within the `VP`.

In order to obtain accurate counts, one must discover such variations and construct patterns or sets of patterns that capture just the desired structures. This frequently requires considerable trial and error. One could, in theory, detect over-generalization by scanning the selected structures by hand to verify that they are of the appropriate type. But this is very time consuming for any structure that occurs more than a few dozen times. Detecting under-generalization is even more difficult—akin to looking for the proverbial needle in a haystack.

One possible technique for solving both problems involves creating two patterns: an under-general one that captures a subset of the desired sentences and an over-general one that captures a superset. By repeatedly examining the difference between these sets and modifying the patterns to bring them closer together, one could hope to eventually reach an accurate common pattern. However, in practice it is hard to produce a pattern that is known to over-generalize, due to the possibility of unexpected variation, and it is hard to reduce the over-general pattern without making mistakes.

The analyses reported here are based on a somewhat different, and apparently novel, approach that largely avoids the problem of detecting under-generalization. This method involves dividing a set of structures into mutually exclusive subsets that completely span the original set. For example, the set of all verb phrases was grossly partitioned into 25 classes, or subsets, and a set of TGREP patterns was created for each. In cases where the classes overlapped, their patterns were modified to eliminate over-generalization. Phrases not classified by any of the patterns were then scanned by eye. If one of these phrases clearly should have fallen into an existing class, the class was expanded appropriately. If an unclassified phrase did not fit into any existing class (and was not deemed a mis-tag), a new class was created. In the end, because the subsets span nearly the whole space of verb phrases and do not overlap, we can be quite confident that they were not over- or under-general.

Each of the subsets can then, recursively, be partitioned if a more fine-grained analysis is needed. Performing the recursive partitioning also helps detect cases where phrases were misclassified at the top level. The drawback of this approach is that one may have to classify many more phrases than one is actually interested in. However, the advantage is that it avoids haystacks almost entirely. That is, one never need look through a large set of good sentences to find a few bad ones, or vice-versa.

A program, known as *dif*, was written to take a set of TGREP pattern files, extract their phrase sets from the corpus, and compare them, producing a report of the number of phrases unique to each set or found in the overlap between pairs of sets. It is also possible to extract the phrases in any of those subsets. This makes it quite easy to check a partitioning for over- and under-generalization. The only phrases that the researcher need look through by eye are those that are not captured by any pattern and those that are captured by multiple patterns. That is, one need not sort through the good phrases to find the bad ones.

Another useful technique made possible by *dif* is to isolate phrases by intersecting sets produced by two or more patterns. For example, one series of patterns might be used to classify the contents of relative clauses: whether they involve subject or object extractions. Another series of patterns could classify the context of relative clauses: whether they modify the matrix subject or object. By intersecting the sets produced by these patterns, one could select the object extraction clauses modifying matrix subjects without the need for writing a special set of patterns for this specific case.

TGREP patterns, even for relatively simple structures, tend to be quite long and complex. Furthermore, due to variation, several patterns are normally required to get all cases of a desired construction. Because there is a lot of overlap among those patterns, a pre-processor was written that allows shared components of patterns to be stored in macros. This reduces errors by simplifying the patterns and, when a change is needed to a common part of the patterns, the macro can be updated, affecting all patterns in a single step.

The experience gained in using TGREP to carry out the majority of this analysis made apparent some of its limitations. Therefore, I designed an improved version of the program, known as TGREP2. Built from scratch, TGREP2 is able to handle all TGREP patterns, but it also has a variety of new features. One major addition is that it allows patterns to contain boolean expressions of operators. For example, one can specify a `VP` that contains a *VBN or* that contains another `VP` which contains a *VBN*. This reduces much of the need for writing multiple patterns to match a single construction. TGREP2 also permits flexible formatting of results. Among other things, this allows phrasal nodes to be identified with a unique code. Using these codes in the *dif* program, rather than the actual phrases, is faster and avoids problems caused by phrases that repeat more than once. TGREP2 is described in more detail in Appendix C.

4.2 Verb phrases

The initial goal of this portion of the study was to determine the frequency of usage of various verb tenses and voices. However, it was quickly determined that tense could not be directly queried in the corpora due to the variety of ways in which verb phrases (VPs) are expressed and tagged. Therefore, a rather large-scale investigation of VPs was necessary. This later proved additionally useful because most of the ambiguous constructions that are of interest in this thesis depend on specific verb argument structures and tenses. Therefore, a complete analysis of verb phrases is a useful first step in studying the more specific constructions.

Table 4.1 shows the major classes of the 180,405 VPs in the Wall Street Journal (WSJ) corpus and the 196,575 VPs in the Brown corpus (BC). "W #" and "B #" indicate the number of phrases of the specified type in the WSJ and BC, respectively. "W %" and "B %" indicate this as a percentage. The vast majority of phrases were placed in a single class. Just 151 (0.08%) of the WSJ phrases and 50 (0.03%) of the BC phrases did not fit into any category. Many of these appear to have been mis-tagged and there is not much regularity in these errors. 55 (0.03%) of the WSJ and 76 (0.04%) of the BC sentences were classified in more than one category. Many of these are due to tagging errors in complex VPs. For example, a sentence like *is eating and running* would ordinarily be encoded:

- `(VP (VBZ is) (VP (VP (VBG eating) (CC and) (VBG running))))`

but was occasionally marked as:

- `(VP (VBZ is) (VP (VBG eating) (CC and) (VBG running)))`

It is important to note that `VP`s are frequently nested within one another. Only 65,341 (36.2%) of the WSJ `VP`s are not contained within another `VP`. Figure 4.1 shows the percentage of `VP`s at each level of nesting one `VP` within another. In cases where nesting occurred, each of the phrases was separately classified. One might wish to analyze

W %	B %	W #	B #	Class	Description	
17.1%	17.4%	30,763	34,132	VB	Infinitives: *eat*, *be outlawed*, *make filters*	
15.4%	14.3%	27,872	28,116	VBD	Simple past tense: *ate*	
10.9%	7.59%	19,690	14,920	VBZ	Simple present tense: *eats*	
11.8%	13.5%	21,319	26,589	VBN	Past participle: *eaten*	
7.95%	7.66%	14,338	15,048	VBG	Present participle or gerund: *eating*	
8.93%	7.64%	16,111	15,012	to	`VP` starting with *to*, usually followed by a `VP`	Table 4.2
8.32%	8.31%	15,003	16,327	is	Any phrase with main verb *is*, *are*, or *am*	Table 4.3
4.04%	6.75%	7,289	13,269	was	Any phrase with main verb *was* or *were*	Table 4.4
4.44%	3.39%	8,009	6,663	has	Any phrase with main verb *has* or *have*	Table 4.5
1.22%	2.68%	2,202	5,268	had	Any phrase with main verb *had*	Table 4.6
2.30%	1.36%	4,144	2,675	will	*will* or *won't*, usually followed by a `VP`	Table 4.7
4.26%	5.82%	7,690	11,446	modal	Any modal verb other than *will* or *won't*	Table 4.8
0.30%	0.43%	543	837	NN	Marked as nouns, actually a variety of types	
0.11%	0.21%	190	421	JJ	Marked as adjectives, actually VBD or VBZ	
0.01%	0.00%	25	0	about-to	*about to* `VP`: *about to find out*	
2.38%	2.63%	4,300	5,178	CONJ	Conjoined verb phrases: *eat and run*	
0.06%	0.27%	109	525	COMMA-VP	`VP`, `VP`: *redistribute wealth, not create it*	
0.04%	0.01%	69	23	COMMA-NP	`VP`, `NP`: *fell 13%, a major decline*	
0.04%	0.01%	67	21	COMMA-S	`VP`, `S`: *fell 13%, which is a major decline*	
0.02%	0.06%	35	118	VP	`VP` that simply contains another `VP`	
0.24%	0.00%	434	0	NONE	`VP` headed by a `-NONE-` trace (WSJ only)	
0.06%	0.00%	111	1	NP	`VP` consisting of just an `NP`, likely errors	
0.02%	0.00%	43	0	ADJP	`VP` consisting of just an adjective phrase	
0.02%	0.00%	32	0	PP	`VP` consisting of just a prepositional phrase	
0.01%	0.01%	24	19	ADVP	`VP` consisting of just an adverb phrase	
10.1%	0.03%	151	50	unclassified	0.03% | 0.04% | 55 | 76 | overclass.	

Table 4.1: Breakdown of the 180,405 verb phrases in the WSJ and the 196,575 verb phrases in the BC.

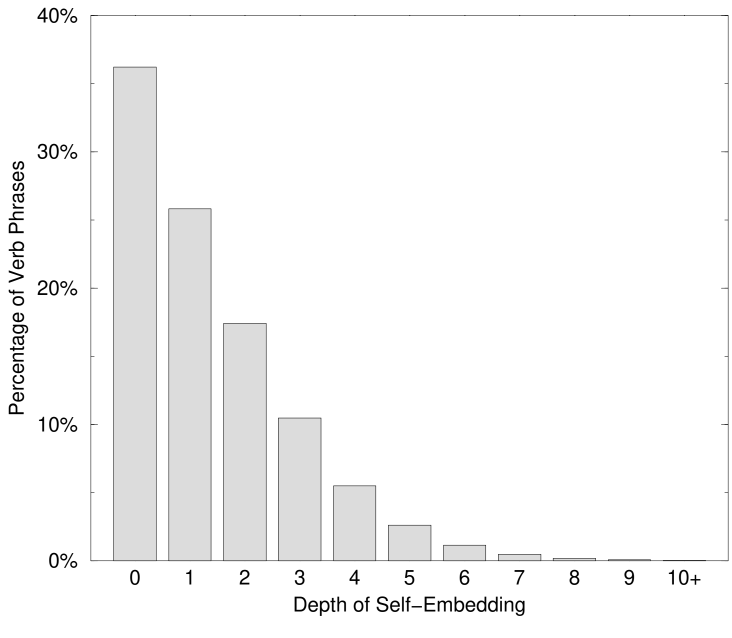


Figure 4.1: Percentage of verb phrases by depth of embedding in the WSJ corpus.

just the top-level phrases, and in fact this could be done quite easily with a simple modification to the patterns used here.

The major phrase classes are as follows:

VB: VPs with a base or infinite verb form, `VB`, as the root, but excluding has phrases. This most common form of VP is not actually a complete phrase. Although phrases of this type, such as *make money*, could be imperatives, that would be very unlikely, especially in the WSJ. Most of them actually appear as complements of *to*, *will*, or another modal. Thus, they are nested within other VPs, such as to-VB, will-VB, or modal-VB.

VBD: Simple past tense verbs, excluding those phrases classified as was or had-VP. These constitute about 15% of VPs, overall.

VBZ: Simple present tense verbs, excluding is and has phrases. These are less common than the simple past tense, particularly in the BC.

VBN: Headed by a past participle. A few cases where *had* was incorrectly classified as a `VBN`, rather than a `VBD`, were excluded. VBN phrases could appear in isolation as reduced relatives, but most of them serve as the passive part of phrases such as is-PAS, was-PAS, or will-be-PAS.

VBG: Headed by a present participle. These too could serve as reduced relatives but are more likely to occur in structures such as is-VBG.

to: Beginning with the word *to*, nearly all of which are followed by a `VP`. A small percentage of these appear as reduced relatives or subordinate clauses, but the majority serve as complements within another `VP`, as in *continued to slide*. Analyzed further in Table 4.2.

is: Use the verbs *is*, *are*, or *am*, which also occur as *Is*, *'s*, *Are*, *'re*, *Am*, or *'m*. *'s* is sometimes incorrectly marked as a possessive (POS). This class includes a variety of structures, such as predicate nominatives, predicate adjectives and *to be* phrases. Analyzed further in Table 4.3.

was: Use the verbs *was* or *were*, including *Was*, *Were*, *wasn't*, etc. Analyzed further in Table 4.4.

has: *has* or *have*, including *Has*, *Have*, *'ve*, and *hath*. Analyzed further in Table 4.5.

had: *had*, *Had*, and *'d*. Analyzed further in Table 4.6.

will: *will* or *won't*, including *Will*, *'ll*, *wo*, and *Wo*. Analyzed further in Table 4.7.

modal: Modal verbs other than those classified as will. Analyzed further in Table 4.8.

NN: VPs with a head marked as a modifying noun. Presumably these are tagging errors since the same words, used in similar ways, are more frequently labeled with verb forms. If treated as verbs, these would be mostly present tense, `VBZ`, forms, with some present participles, `VBG`, and a few past tenses, `VBD`.

JJ: VPs with a head marked as an adjective. Again, these are most likely errors, with the correct forms being mainly `VBD` with a few `VBG` and `VBZ`.

about-to: A set of phrases with the structure `(VP (IN about) (S (VP (TO to)...)))`. *About* is sometimes also tagged as an adverb, `RB`. *About* appears to be the only word, other than a verb or a modal, that can act in this way, an interesting feature of English.

CONJ: A VP containing two or more `VP`s connected by a conjunction, usually *and*. These are the greatest source of over-classification, or inclusion of the same VP in more than one category, because they are frequently mis-tagged, as mentioned earlier.

COMMA-VP: Similar to a CONJ, except that no conjunction is used. For example, *keep the ball down, move it around,* or *saw the film Monday, ran for 30 minutes, then went in*.

COMMA-NP: A `VP` followed by a comma and a modifying noun phrase. For example, *raised its quarterly dividend, a change that will increase the...*

W %	B %	W #	B #	Class	Description
100%	99.9%	16,109	14,999	to-VP	*to* followed by a `VP`
98.7%	88.5%	14,744	13,213	to-VB	*to eat*
8.91%	11.5%	1,330	1,718	to-be	*to be...*
51.9%	47.8%	690	821	to-be-VP	*to be* `VP`
85.5%	89.6%	590	736	to-be-PAS	*to be eaten*
14.3%	6.46%	99	53	to-be-VBG	*to be eating*
0.14%	3.90%	1	32	to-be-NONE	*to be []*
0.00%	0.24%	0	2	unclassified	0.00% | 0.37% | 0 | 3 | overclass.
20.9%	23.7%	278	408	to-be-NP	*to be the next CEO*
19.6%	21.3%	261	366	to-be-ADJ	*to be hungry*
4.81%	4.48%	64	77	to-be-PP	*to be in a general retreat*
2.11%	0.81%	28	14	to-be-ADV	*to be back at work by today*
0.60%	0.52%	8	9	to-be-S	*to be how fast Kasparov could win game two*
0.08%	1.51%	1	26	unclassified	0.00% | 0.12% | 0 | 2 | overclass.
0.20%	0.66%	30	98	VP-to-NONE	*to* followed by nothing or a null VP
0.09%	0.14%	14	21	unclassified	0.06% | 0.13% | 9 | 19 | overclass.
0.01%	0.09%	2	13	unclassified	0.00% | 0.00% | 0 | 0 | overclass.

Table 4.2: Sub-classes of the 16,111 WSJ and 15,012 BC to phrases.

COMMA-S: A `VP` followed by a comma and a modifying relative clause. For example, *delivered by January, which would be a strain in most years*.

VP: Simply a `VP` that contains nothing but another `VP`, as in `(VP (VP ...))`. Their rarity indicates that these may be coding errors.

NONE: A `VP` with a missing head, marked as `-NONE-`. These occur in structures such as is-NONE or was-NONE (see below), which are themselves part of a parallel or gapped construction like, *He spends his days sketching passers-by, or trying to* `[(VP (-NONE))]`. These are only used in the new tagging style and thus do not occur in the BC.

NP: A `VP` that has no verb but contains an `NP`. These tend to arise in gapped constructions, such as, *suspended its common distribution in June 1988 and the distribution on its cumulative convertible acquisition preferred units in September.* In this case, *the distribution on its...* is encoded `(VP (NP=3 (NP (DT the) (NN distribution))...))`. It seems that using a `(VP (-NONE))` would be more appropriate in that situation, but perhaps there are reasons for the different encoding.

ADJP: Similar to NP but with an adjective phrase.

PP: Similar to NP but with a prepositional phrase.

ADVP: Similar to NP but with an adverb phrase.

4.2.1 Verb phrase sub-classes

This section discusses the breakdown of verb phrases in seven of the main categories, to, is, was, has, had, will, and modal, which are shown in Tables 4.2–4.8. Indented rows in the tables indicate subdivisions of the preceding class. For example, Table 4.2 shows that to has a single subclass, to-VP, which accounts for 100% of the WSJ and 99.9% of the BC to phrases. Subclass to-VP is subdivided into to-VB, to-be, and to-NONE. Together these account for all but 14 of the 16,109 WSJ phrases and overlap on 9 of them. In the BC, these account for all but 21 of the 14,933 to-VP phrases and overlap on 19 of them. Subclass to-be is further divided into to-be-VP, to-be-NP, to-be-ADJ, to-be-PP, to-be-ADV, and to-be-S, and so on.

to-VP: *to* followed by a `VP`, which is the only grammatical option.

W %	B %	W #	B #	Class	Description
39.9%	35.0%	5,993	5,715	is-VP	*is* followed by a `VP`
48.3%	64.8%	2,892	3,704	is-PAS	*is* followed by a passive construction
7.02%	3.35%	203	124	is-being-PAS	*is being* or *is getting* followed by a `VP`
50.5%	21.3%	3,027	1,218	is-VBG	*is* followed by a present participle: *is eating*
0.80%	14.1%	48	804	is-NONE	*is* with a missing `VP`
0.20%	0.17%	12	10	is-about-to	*is about to* followed by a `VP`
0.20%	0.23%	12	13	is-VB	*is get off the screen*
0.13%	0.05%	8	3	unclassified	0.08% 0.40% 5 23 overclass.
25.2%	30.5%	3,784	4,981	is-NP	Noun phrase complement
23.0%	23.8%	3,452	3,897	is-ADJ	Adjective phrase complement
5.66%	5.34%	849	872	is-PP	*is in preliminary stages*
1.73%	0.78%	259	128	is-ADV	*is up 25%*
4.54%	4.92%	681	804	is-S	*is that the U.S. trade law is working*
0.17%	0.13%	26	22	unclassified	0.31% 0.58% 47 94 overclass.

Table 4.3: Sub-classes of the 15,003 WSJ and 16,327 BC is phrases.

to-VB: *to* followed by a verb infinitive such as, *to eat*. Infinitive forms were usually encoded as `VB`, but are occasionally labeled nouns, `NN`, or adjectives, `JJ`. *to be* was excluded.

to-be: *to be*, quite literally.

to-be-VP: *to be* followed by any VP.

to-be-PAS: *to be* followed by a passive construction. Ordinarily this was headed by a `VBN`, but was occasionally marked with a `VBD`, `JJ`, or `NN`. Note that the passive is the most common use of *to be*.

to-be-VBG: *to be* followed by a present participle.

to-be-NONE: *to be* followed by a null phrase, usually because this is part of a gapped construction or a question. In the WSJ, these are known to be VPs. However, in the BC, the type of the phrase trace is not given. Because of the higher frequency of questions in the BC, it is likely that many of the missing phrases were noun phrases or other non-verb phrases. However, for the purpose of this study, they were all assumed to be VPs.

to-be-ADJ: *to be* followed by an adjectival predicate.

to-be-PP: *to be* followed by a prepositional phrase.

to-be-ADV: *to be* followed by an adverbial.

to-be-S: *to be* followed by a sentential.

to-NONE: *to* followed by a null `VP`, due to a gapped or shifted construction.

Table 4.3 contains a breakdown of the is phrases. These all begin with either *is* or *are* or their contracted forms: *'s* or *'re*. The relatively large number of over-classified phrases here and in similar constructions is mainly due to conjoined phrases, which do reasonably satisfy more than one category. For example, *is running and scared* would be classified as both an is-VBG and an is-ADJ.

The subclasses are as follows:

is-VP: *is* followed by any VP.

is-PAS: *is* followed by a past participle, which could be marked `VBN`, `VBD`, or `JJ`. Also includes *being* or *getting* (marked as `VBG`) followed by a `VP`.

is-being-PAS: *is being* or *is getting* followed by a `VP`.

W %	B %	W #	B #	Class	Description
49.2%	43.4%	3,588	5,765	was-VP	*was* followed by a VP
81.7%	67.8%	2,931	3,909	was-PAS	*was eaten*
1.33%	1.56%	39	61	was-being-PAS	*was being eaten*
17.8%	23.9%	639	1,378	was-VBG	*was eating*
1.17%	8.48%	42	489	was-NONE	*…wasn't eating but I was []*
0.20%	0.33%	7	19	was-about-to	*was about to* VP
0.08%	0.12%	3	7	was-VB	*all I could do was eat*
0.08%	0.10%	3	6	unclassified	0.08% 0.17% 3 10 overclass.
20.1%	26.5%	1,468	3,513	was-NP	Noun phrase complement
19.0%	19.4%	1,388	2,574	was-ADJ	Adjective phrase complement
5.79%	6.41%	422	850	was-PP	*was under the table*
3.92%	1.26%	286	167	was-ADV	*was down*
2.10%	3.59%	153	477	was-S	*was to leave China today*
0.04%	0.11%	3	14	unclassified	0.26% 0.71% 19 94 overclass.

Table 4.4: Sub-classes of the 7,289 WSJ and 13,269 BC was phrases.

is-VBG: *is* followed by a present participle or present participle marked as an adverb. This excludes cases where the present participle is *being* or *getting* and is followed by a VP, as those form passives.

is-NONE: *is* with a null VP.

is-about-to: *is about to*, which seems to be invariably followed by a VP.

is-VB: *is* followed by a root-form verb. These almost always follow an expression like, *all they need to do…,* or, *the first thing to do….*

is-NP: A predicate nominative.

is-ADJ: A predicate adjective.

is-PP: *is* with a prepositional phrase complement. For example, *aren't off the hook,* or, *are on the mark*.

is-ADV: *is* with an adverbial complement. These usually involve the adverbs *up*, *down*, and *out*.

is-S: *is* followed by a sentential complement or quotation.

Table 4.4 gives the breakdown of the was phrases, which are headed by *was* or *were*. Tables 4.5 and 4.6 analyze the has and had classes, Table 4.7 covers the will phrases, and Table 4.8 the other modal verbs. The subclasses are analogous to those for is.

That there is quite a bit of shared structure in the classes shown in Tables 4.2–4.8 is a testament to the composability of English. Each class, or one of its subclasses, can be divided into -VP, -NP, -ADJ, -PP, -ADV, or -S forms. The -VP phrases also break down into -PAS or -VBG. The -NONE phrases are forms of either -PAS or -VBG in the WSJ and forms of any of the seven in the BC.

This consistency lends support to the idea that English could be modeled by a context-free grammar. In such a model, VPs starting with *to be*, *is*, *was*, *has been*, *had been*, *will be*, or *should be* would all lead to the same non-terminal state. However, would such a model still be appropriate in the stochastic domain in which probabilities are given for each context-free production? This would be an accurate model only if the relative frequencies of the sub-classes were roughly equivalent across these classes.

Table 4.9 shows the frequencies of the six major subdivisions of to-be, is, was, has-been, had-been, will-be, and modal-be, and the two major subdivisions of the -VP phrases. The values actually seem reasonably consistent across the upper set of phrase types. It would not be terribly inaccurate to model these with a single non-terminal symbol that produced the six subclasses with fixed probabilities.

The comparison of -PAS to -VBG, however, is less consistent. The ratio in their frequencies ranges from 9:1 for will-be- and modal-be- to 1:1 for is- in the WSJ. Interestingly, has-been and had-been are in reasonably close agreement, but

W %	B %	W #	B #	Class	Description
72.1%	66.9%	5,772	4,455	has-VP	*has* followed by a `VP`
72.9%	67.2%	4,207	2,994	has-VBN	*has eaten*
26.5%	34.5%	1,532	1,536	has-been	*has-been…*
61.7%	60.5%	945	929	has-been-VP	*has-been* `VP`
65.7%	83.7%	621	778	has-been-PAS	*has been eaten*
34.2%	12.8%	323	119	has-been-VBG	*has been eating*
0.21%	3.66%	2	34	has-been-NONE	*has-been []*
0.21%	0.11%	2	1	unclassified	0.11% 0.32% 1 3 overclass.
14.6%	16.0%	223	246	has-been-NP	*have been good returns*
17.3%	11.9%	265	183	has-been-ADJ	*has been sluggish*
4.70%	5.21%	72	80	has-been-PP	*have been at odds*
0.85%	0.85%	13	13	has-been-ADV	*has been off recently*
0.91%	0.85%	14	13	has-been-S	*has been to raise prices*
0.07%	0.39%	1	6	unclassified	0.07% 0.20% 1 3 overclass.
0.59%	1.55%	34	69	has-NONE	*did not go as far as he could have []*
0.02%	0.00%	1	0	unclassified	0.02% 0.09% 1 4 overclass.
22.3%	25.0%	1,784	1,667	has-NP	*has* `NP`
0.04%	0.14%	3	9	has-ADJ	*has* ADJ (likely mis-tagged)
0.00%	0.20%	0	13	has-PP	*has* `PP` (likely mis-tagged)
5.74%	6.66%	460	444	has-S	*have to worry whether the ad is truthful*
0.02%	0.23%	2	15	unclassified	0.17% 0.14% 14 9 overclass.

Table 4.5: Sub-classes of the 8,009 WSJ and 6,663 BC has phrases.

W %	B %	W #	B #	Class	Description
60.3%	71.3%	1,327	3,757	had-VP	*had* followed by a `VP`
71.3%	76.0%	946	2,856	had-VBN	*had eaten*
28.4%	25.6%	377	960	had-been	*had-been…*
67.4%	56.9%	254	546	had-been-VP	*had-been* `VP`
67.7%	76.4%	172	417	had-been-PAS	*had been eaten*
32.7%	19.0%	83	104	had-been-VBG	*had been eating*
0.00%	4.95%	0	27	had-been-NONE	*had-been []*
0.00%	0.00%	0	0	unclassified	0.39% 0.37% 1 2 overclass.
15.9%	17.7%	60	170	had-been-NP	*had been good returns*
8.49%	10.6%	32	102	had-been-ADJ	*had been sluggish*
5.31%	7.81%	20	75	had-been-PP	*had been at odds*
2.12%	0.94%	8	9	had-been-ADV	*had been off recently*
0.53%	0.83%	2	8	had-been-S	*had been to raise prices*
0.27%	0.10%	1	1	unclassified	0.00% 0.31% 0 3 overclass.
0.38%	1.38%	5	52	had-NONE	*did better than I had [] on the test*
0.00%	0.00%	0	0	unclassified	0.08% 0.11% 1 4 overclass.
35.0%	20.7%	770	1,091	had-NP	*had a package*
0.00%	0.06%	0	3	had-ADJ	Coding error
0.00%	0.27%	0	14	had-PP	*had at any time before*
5.18%	6.45%	114	340	had-S	*had to spend the day there*
0.00%	0.38%	0	20	unclassified	0.41% 0.17% 9 9 overclass.

Table 4.6: Sub-classes of the 2,202 WSJ and 5,268 BC had phrases.

W %	B %	W #	B #	Class	Description				
99.8%	99.6%	4,136	2,664	will-VP	*will* followed by a `VP`				
72.2%	68.8%	2,988	1,833	will-VB	*will eat*				
0.27%	1.04%	8	19	will-have-VP	*will have* `VP`				
87.5%	84.2%	7	16	will-have-VBN	*will have eaten*				
12.5%	0.00%	1	0	will-have-been-PAS	*will have been eaten*				
27.4%	30.1%	1,132	802	will-be	*will be...*				
54.4%	47.0%	616	377	will-be-VP	*will be* `VP`				
89.6%	85.4%	552	322	will-be-PAS	*will be eaten*				
10.4%	10.9%	64	41	will-be-VBG	*will be eating*				
0.49%	1.33%	3	5	will-be-NONE	*you may not be eating, but I will be []*				
0.00%	2.39%	0	9	unclassified	0.00%	0.00%	0	0	overclass.
15.8%	21.1%	179	169	will-be-NP	*will be the recipient*				
22.1%	20.4%	250	164	will-be-ADJ	*will be hungry*				
4.24%	7.48%	48	60	will-be-PP	*will be on television*				
1.77%	1.62%	20	13	will-be-ADV	*will be down*				
1.06%	1.37%	12	11	will-be-S	*will be to spend more money*				
0.62%	1.87%	7	15	unclassified	0.00%	0.37%	0	3	overclass.
0.48%	1.05%	20	28	will-NONE	*you may not be eating, but I will []*				
0.05%	0.30%	2	8	unclassified	0.15%	0.26%	6	7	overclass.
0.02%	0.41%	1	11	will-NP	*will you*?				
0.05%	0.00%	2	0	unclassified	0.00%	0.00%	0	0	overclass.

Table 4.7: Sub-classes of the 4,144 WSJ and 2,675 BC will phrases.

W %	B %	W #	B #	Class	Description				
72.3%	65.5%	5,561	7,498	modal-VB	*would remain*				
0.14%	0.89%	11	102	modal-VBN	These appear to be tagging mistakes				
0.13%	0.17%	10	19	modal-VBP	These appear to be tagging mistakes				
0.10%	0.45%	8	52	modal-VBD	These appear to be tagging mistakes				
26.0%	31.0%	1,996	3,544	modal-be	*may be...*				
54.4%	60.1%	1,085	2,129	modal-be-VP	*should be* `VP`				
91.3%	92.6%	991	1,972	modal-be-PAS	*could be eaten*				
8.11%	5.07%	88	108	modal-be-VBG	*might be eating*				
0.37%	2.16%	4	46	modal-be-NONE	*but I should be []*				
0.18%	0.14%	2	3	unclassified	0.00%	0.28%	0	6	overclass.
16.1%	14.1%	321	498	modal-be-NP	*could be a movie star*				
23.5%	19.8%	469	703	modal-be-ADJ	*can be expensive*				
2.91%	3.24%	58	115	modal-be-PP	*may be on the dumb side*				
1.20%	1.13%	24	40	modal-be-ADV	*wouldn't be here*				
1.90%	1.52%	38	54	modal-be-S	*would be to sell the shipyard to an outsider*				
0.05%	0.59%	1	21	unclassified	0.00%	0.20%	0	7	overclass.
0.07%	0.37%	5	42	modal-NP	*but I can't []*				
0.04%	0.10%	3	12	modal-PP	*but I can't []*				
0.78%	1.58%	60	181	modal-NONE	*but I can't []*				
0.49%	0.46%	38	53	ought-to	The one that doesn't fit the modal mold				
0.21%	0.44%	16	50	unclassified	0.21%	0.36%	16	41	overclass.

Table 4.8: Sub-classes of the 7,690 WSJ and 11,446 BC modal phrases.

	to-be-		is-		was-		has-been-		had-been-		will-be-		modal-be-	
	WSJ	BC	WSJ	BC	WSJ	BC	WSJ	BC	WSJ	BC	WSJ	BC	WSJ	BC
-NP	20.9%	23.7%	25.2%	30.5%	20.1%	26.5%	14.6%	16.0%	15.9%	17.7%	15.8%	21.1%	16.1%	14.1%
-ADJ	19.6%	21.3%	23.0%	23.8%	19.0%	19.4%	17.3%	11.9%	8.49%	10.6%	22.1%	20.4%	23.5%	19.8%
-PP	4.81%	4.48%	5.66%	5.34%	5.79%	6.41%	4.70%	5.21%	5.31%	7.81%	4.24%	7.48%	2.91%	3.24%
-ADV	2.11%	0.81%	1.73%	0.78%	3.92%	1.26%	0.85%	0.85%	2.12%	0.94%	1.77%	1.62%	1.20%	1.13%
-S	0.60%	0.52%	4.54%	4.92%	2.10%	3.59%	0.91%	0.85%	0.53%	0.83%	1.06%	1.37%	1.90%	1.52%
-VP	51.9%	47.8%	39.9%	35.0%	49.2%	43.4%	61.7%	60.5%	67.4%	56.9%	54.4%	47.0%	54.4%	60.1%
-PAS	85.5%	89.6%	48.3%	64.8%	81.7%	67.8%	65.7%	83.7%	67.7%	76.4%	89.6%	85.4%	91.3%	92.6%
-VBG	14.3%	6.46%	50.5%	21.3%	17.8%	23.9%	34.2%	12.8%	32.7%	19.0%	10.4%	10.9%	8.11%	5.07%

Table 4.9: Frequencies of major subdivisions of simple verb phrases

is and was produce quite different distributions in the WSJ but not in the BC. This suggests that a stochastic context-free model of English may need to represent these structures with non-terminal states differing in their production frequencies.

4.2.2 Tense and voice

Finally, we now have nearly all the data needed to return to the original question of interest, which was the frequency of the verb tenses and voices. It is common to categorize English verbs into twelve tenses based on whether they are in present, past, or future time, and simple, perfect, progressive, or perfect progressive aspect. These classes can be further divided into active and passive voices. Examples of these phrase types, along with their frequencies, are shown in Table 4.10. Because the construction is so rare, the analyses of the *is going to* phrases were not shown in the earlier tables. Forms of the verb *to be* followed by noun, adjective, or prepositional phrases or other complements were also excluded, as were infinitives, modals, and VP compounds.

Overall, 67,411 WSJ phrases and 63,008 BC phrases were considered in this analysis. Respectively, 89.0% and 85.2% of them are in the active voice. As should be expected, the simple aspect is dominant, accounting for close to 85% of both actives and passives in both corpora. Perfect aspect accounts for between 8.6% and 12.8% of the phrases and progressive aspect for 5 to 6% of the actives and 2 to 3% of the passives. The perfect progressive aspect is rarely used in the active and never in the passive.

In the active voice, the past tense is somewhat more common than the present in the WSJ corpus and considerably more common than the present in the BC. This is perhaps a surprising result as one might expect the WSJ, consisting entirely of newspaper articles, to be heavily weighted toward the past tense. In the passive voice, the present tense is actually a bit more common than the past in both corpora. Thus, in terms of frequency, there is some interaction between voice and tense.

4.2.3 Comparison of the WSJ and BC

Because the WSJ has been tagged in the newer Treebank II style, while the BC remains in the older Treebank I style, significantly different TGREP patterns were needed to extract phrases from them. A principal difference is that, in the BC, auxiliary verbs including modals and the verb *to be* are usually, though not always, encoded as a separate `AUX` prior to or in place of the `VP`. The Treebank II style is more consistent in enclosing all verb phrase components within the `VP`.

Despite the differences in content and coding style, there is a remarkable consistency in VP usage between the corpora, as evidenced in the preceding tables. However, there are some differences worth noting. The BC has many more `-NONE-` constructions, which are in some cases as much as 10-20 times more frequent than in the WSJ. One reason for this is that the `-NONE-` is a trace marker which often occurs in questions, which are rare in the WSJ. In the WSJ, the type of the trace is marked and only VPs were counted. But in the BC the type of, or even the existence of, the trace was not marked. It is difficult to determine automatically the types of these traces. Therefore, for the purpose of this study, they were all assumed to be VPs, although it is likely that only about half of them were actually VPs.

Active Voice	Present		Past		Future		Total	
	WSJ	BC	WSJ	BC	WSJ	BC	WSJ	BC
Simple	eats		ate		will/is going to eat		84.5%	83.7%
	19,690	14,920	27,872	28,116	3,131	1,924	50,693	44,960
Perfect	has eaten		had eaten		will/i.g.t. have eaten		8.60%	10.9%
	4,207	2,994	946	2,856	7	16	5,160	5,866
Progressive	is eating		was eating		will/i.g.t be eating		6.22%	4.91%
	3,027	1,218	639	1,378	68	41	3,734	2,637
Perf. Prog.	has been eating		had been eating		will/i.g.t. have been eating		0.68%	0.42%
	323	119	83	104	0	0	406	223
Total	45.4%	35.9%	49.2%	60.5%	5.34%	3.69%	89.0%	85.2%
	27,247	19,251	29,540	32,454	3,206	1,981	59,993	53,686
Passive Voice	Present		Past		Future		Total	
	WSJ	BC	WSJ	BC	WSJ	BC	WSJ	BC
Simple	is eaten		was eaten		will/i.g.t be eaten		86.0%	85.2%
	2,892	3,704	2,931	3,909	559	329	6,382	7,942
Perfect	has been eaten		had been eaten		will/i.g.t. have been eaten		10.7%	12.8%
	621	778	172	417	1	0	794	1,195
Progressive	is being eaten		was being eaten		will/i.g.t be being eaten		3.26%	1.98%
	203	124	39	61	0	0	242	185
Perf. Prog.	has been being eaten		had been being eaten		will/i.g.t. have been being eaten		0.00%	0.00%
	0	0	0	0	0	0	0	0
Total	50.1%	49.4%	42.4%	47.1%	7.55%	3.53%	11.0%	14.8%
	3,716	4,606	3,142	4,387	560	329	7,418	9,322

Table 4.10: Frequencies of verb tenses

As mentioned earlier, there are some interesting differences in the use of verb tense in the two corpora. As shown in Table 4.10, the WSJ slightly favors the past tense in the active voice but the BC strongly favors it. In the passive voice, the WSJ favors the present tense while the BC favors it by a narrower margin. Thus, the BC material makes more use of the past tense, which probably reflects the influence of the fictional content. Overall, the future tense is used rather infrequently and is more common in the WSJ. The passive voice is used more in the BC, accounting for nearly 15% of the phrases, compared to the WSJ's 11%.

4.3 Relative clauses

The analysis of relative clauses (RCs) is somewhat more complex than that of VPs. In this case, we are interested not only in the contents of the clauses but in their locations as well. In particular, we would like to know if the distribution of RC types is dependent on the role (subject or object) of the noun phrase being modified. This was determined by separately classifying the RCs on the bases of contents and context and then intersecting those classes. We also need to perform separate analyses on reduced RCs—those without an explicit relative pronoun—as they are encoded using rather different syntax. Finally, RCs are not encoded in the same manner in the Treebank I and II bracketing styles so the WSJ and BC must be treated quite differently.

4.3.1 Simple relative clause types

The first dimension on which RCs were categorized was whether they were introduced by an explicit relative pronoun (*the boy <u>who</u> was running home…*) or whether they were reduced (*the boy [] running home…*), as these two forms were generally encoded using quite different syntactic structures.

Table 4.11 contains the major taxonomy of marked (unreduced) relative clauses and Table 4.12 analyzes the VPs within these same clauses. The analysis of the VP is necessary to distinguish among some forms of subject-relative, object-relative, and passive constructions. A summary of the results is shown in Tables 4.14 and 4.15, discussed at the

W %	B %	W #	B #	Class	Description
87.9%	77.1%	7,021	6,058	RC-WHN	Noun phrase extracted RC
91.5%	90.2%	6,422	5,465	RC-WHN-NS	RC with no subject phrase
88.6%	87.0%	5,692	4,755	subject-extraction	Subject-extraction RC
11.0%	12.3%	705	672	passive-extraction	Passive RC.
0.25%	0.37%	16	20	RC-to-subject	*who is to run*
0.14%	0.33%	9	18	RC-to-passive	*which is to be chased*
8.25%	10.9%	577	659	RC-WHN-S	Object-extraction RC
0.35%	0.30%	2	2	RC-to-object	*that Congress is to consider*
7.68%	10.6%	613	832	RC-WHA	Adverb phrase extracted RC
4.38%	12.4%	350	974	RC-WHP	Prepositional phrase extracted RC

Table 4.11: Breakdown of the marked relative clauses: 7,983 from the WSJ and 7,862 from the BC.

end of this section.

The marked RC types are as follows:

RC-WHN: RCs in which the modified noun has been extracted from within a noun phrase in the RC.

RC-WHN-NS: Noun-phrase extracted RCs which have a null subject. These include subject-extraction, *the boy who chased the dog*, and passive, *the boy who was chased*, constructions.

subject-extraction: Basic subject-extraction relative clauses. This set is formed from all members of RC-WHN-NS that were not otherwise classified as passive-extraction, RC-to-subject, or RC-to-passive.

passive-extraction: Formed by intersecting RC-WHN-NS with the sets of RCs with passive noun phrases, as shown in Table 4.12.

RC-to-subject: Subject-extraction clauses whose VPs use *to* followed by the infinitive. Although infrequent in the marked form, these are classified separately because their common reduced forms, see below, have properties that distinguish them from other RCs.

RC-to-passive: Passive RCs whose VPs use *to be* followed by past participle. See RC-to-subject.

RC-WHN-S: Noun-phrase extracted RCs with intact subject noun phrases. These are object-extractions.

RC-WHA: RCs formed by extraction from within an adverbial phrase. These tend to use the adverbs *where*, *when*, and sometimes *that*. For example, *instances where the banks made money*, *an era when crime is rampant*, *each day that Congress fails to act*.

RC-WHP: RCs formed by extraction from within a prepositional phrase. These are typically introduced by constructions such as *with which*, *in which*, *of which*, *for which*, or *under whose*.

The analysis of reduced relatives requires a very different set of patterns. These are actually somewhat easier as we do not need to separately classify the VPs to determine the clause type. The main results are shown in Table 4.13. A number of the reduced-relative types, identifiable in the WSJ, either did not occur or could not be detected due to coding differences in the BC.

The reduced RC types are as follows:

RRC-PAS: Classic reduced relatives, such as *the evidence examined by the lawyer....* In this case, both the relative pronoun and the verb *to be* are removed.

RRC-VBG: Subject-extraction reduced relatives formed with the present participle: *the boy running away was lost*.

RRC-WHN: Other noun-phrase extraction RCs.

RRC-OR: Reduced object-relatives.

W %	B %	W #	B #	Class	Description
29.9%	23.9%	2,388	1,863	RC-VBZ	*who study diseases*
24.5%	25.5%	1,955	1,987	RC-VBD	*that developed the cure*
13.3%	14.3%	1,060	1,113	RC-is	*who is...*
33.1%	35.7%	351	397	RC-is-PAS	*which is expected shortly, who is being sued*
2.17%	2.34%	23	26	RC-is-to	*that is to follow*
30.4%	50.0%	7	13	RC-is-to-PAS	*which is to be launched*
6.38%	10.3%	509	804	RC-was	*who was directly involved*
51.9%	39.3%	264	316	RC-was-PAS	*which was sold, who were being attacked*
1.96%	3.23%	10	26	RC-was-to	*who were to own 75%*
40.0%	38.5%	4	10	RC-was-to-PAS	*that was to be delivered*
10.1%	7.62%	807	594	RC-has	*who has studied coaching*
9.79%	11.4%	79	68	RC-has-bn-PAS	*who have been forced to retire*
3.55%	7.49%	283	584	RC-had	*that had reported to her predecessor*
12.4%	14.4%	35	84	RC-had-bn-PAS	*which had been widely expected*
3.61%	2.25%	288	175	RC-will	*who will get letters*
22.2%	13.7%	64	24	RC-will-be-PAS	*which will be restated*
9.31%	11.4%	743	887	RC-MD	*that could result in death*
16.7%	22.9%	124	203	RC-MD-be-PAS	*that can be cross-pollinated*

Table 4.12: Analysis of the VPs within relative clauses. This is used primarily to determine whether relative clauses are active or passive.

W %	B %	W #	B #	Class	Description
48.4%	38.7%	4,135	3,549	RRC-PAS	Classic reduced relatives
20.9%	20.3%	1,787	1,862	RRC-VBG	Reduced subject-extractions
22.6%	36.2	1,931	3,322	RRC-WHN	Other NP-extraction reduced relatives
56.4%	41.1%	1,089	1,367	RRC-OR	Reduced object-extractions
43.3%	58.9%	837	1,955	RRC-to	Noun-modifying clauses starting with *to*
70.3%	86.9%	588	1,698	RRC-to-subject?	*candidates to head the committee*
14.7%	9.36%	123	183	RRC-to-PAS	*grain to be shipped*
12.4%	3.79%	104	74	RRC-to-OR	*a blind girl to cure, many things to do*
2.63%		22		RRC-to-PP	*firm ground to build on*
8.10%	5.12%	692	469	RRC-WHA	*the way he wants to die*
0.22%		19		RRC-NP	*the dog, a classic example of the breed*
0.27%		23		RRC-PP	*those still under GASB rules*
0.54%		46		RRC-NONE	*found a dog [] yesterday, lost for days*

Table 4.13: Analysis of reduced relative clauses: 8,545 from the WSJ and 9,169 from the BC.

Extraction Type	Marked		Reduced		Total	
Subject	36.3%	5,692	11.4%	1,787	47.8%	7,479
Passive	4.5%	705	26.4%	4,135	30.9%	4,840
Object	3.7%	577	7.0%	1,089	10.6%	1,664
Adverbial	3.9%	613	4.4%	692	8.3%	1,305
Prep. Phrase	2.2%	350	0.2%	23	2.4%	373
Total	50.7%	7,935	49.3%	7,726		15,661

Table 4.14: Overall relative clause frequencies in the WSJ.

Extraction Type	Marked		Reduced		Total	
Subject	31.4%	4,755	12.3%	1,862	43.7%	6,617
Passive	4.4%	672	23.5%	3,554	27.9%	4,226
Object	4.3%	659	9.0%	1,367	13.4%	2,024
Adverbial	5.5%	832	3.1%	469	8.6%	1,301
Prep. Phrase	6.4%	974	0.0%	0	6.4%	974
Total	52.1%	7,890	47.9%	7,252		15,142

Table 4.15: Overall relative clause frequencies in the BC.

RRC-to: Includes any noun-modifying clauses introduced by the word *to*.

RRC-to-subject?: Composed of any RRC-to clauses that could not otherwise be classified as RRC-to-PAS, RRC-to-OR, or RRC-to-PP. Unfortunately, this is a heterogeneous set containing a number of subtly distinct types of clauses. Some of these phrases are actually subject-extractions, such as *a truck to transport the gas*. Others describe the content or purpose of an abstract noun. For example, *the decision to transport the gas*.

However, these cases appear to be syntactically indistinguishable and can, in fact, be ambiguous. Consider the phrase, *the plans to leave tomorrow*. This could either refer to plans that will themselves be put on a plane and sent out of the country or to plans indicating that someone is going to be leaving tomorrow. The proper analysis can only be made with the help of context.

RRC-to-PAS: Passive modifiers introduced by to, such as *grain to be shipped*.

RRC-to-OR: A sort of object-extraction RC, such as *a blind girl to cure* or *something to grow*. Many of these are of the form *something/nothing/anything to do*.

RRC-to-PP: Prepositional-phrase extractions with dangling prepositions, such as *a stick to beat them with* or *firm ground to build on*. These are only identifiable in the WSJ.

RRC-WHA: A diverse set of clause types encoded as adverbial extractions. Many of them modify either *the way* or *the time*. Others seem to be identical to some of the clauses that fall into the RRC-to-subject? class.

RRC-NP: Simply another noun phrase that serves to clarify or provide additional information about the modified phrase. For example, *John Doe, the magazine's co-editor*, or *CIA chief William Webster, hardly a Washington malcontent*. These are only identifiable in the WSJ.

RRC-PP: Reduced prepositional phrase extractions. For example, *rude brats, in therapy by age five* or *the measure before the conference yesterday*.

RRC-NONE: Traces left behind by an RC that has been moved, as in *found a dog [] yesterday, lost for days*.

The overall RC statistics for the WSJ and BC are summarized in Tables 4.14 and 4.15. The ∗-to- clauses have been removed because they may be of less interest to most researchers and because of the difficulty of accurate classification.

NP Location	Unmodified		PP-modified		RC-modified		Both	
All NPs	83.3%	671,451	12.5%	100,894	3.9%	31,596	0.2%	1,955
Matrix Subject	86.2%	119,519	10.1%	14,037	3.5%	4,827	0.2%	249
Matrix Object	63.8%	27,515	25.7%	11,090	9.6%	4,162	0.9%	393
Nested NPs	87.3%	100,653	10.2%	11,771	2.4%	2,707	0.1%	115

Table 4.16: Frequency of modification by PPs and RCs based on NP location.

Extraction Type	Marked		Reduced		Total	
Subject-Intransitive	15.7%	4,225	5.5%	1,472	21.1%	5,697
Subject-Transitive	22.0%	5,945	7.8%	2,114	29.9%	8,059
Subject-Sent. Comp.	1.2%	317	0.4%	98	1.5%	415
Passive	5.2%	1,412	28.5%	7,693	33.8%	9,105
Object	4.6%	1,236	9.1%	2,456	13.7%	3,692
Total	48.7%	13,135	51.3%	13,833		26,968

Table 4.17: Overall distribution of selected RC types across both corpora.

Overall, the RC statistics of the WSJ and BC match quite well, the main difference being that the WSJ has a somewhat higher percentage of subject-extractions and passives and the BC has a greater proportion of prepositional-phrase extractions. Reduced clauses account for very nearly half of the WSJ clauses and just under half of the BC clauses.

Marked clauses are dominated by subject-extractions, but subject-extractions appear in the reduced form only about 25% of the time. Perhaps surprisingly, passive constructions, which are often ignored in discussions of comprehensibility, are significantly more common than object-relatives. Passives are reduced about 85% of the time and account for over half of all reduced clauses. Object relatives are about twice as likely to be reduced as marked.

4.3.2 Relative clause context

The RCs were then classified by the location of the modified noun phrase. Of the 7,983 marked RCs in the WSJ, 18.4% modified a matrix subject and 25.2% modified noun phrases located within VPs. The latter noun phrases were typically either direct or indirect objects. Of the 8,545 reduced relatives in the WSJ, 15.1% modified subjects and 30.3% modified objects. In the BC, the marked relative proportions were 13.0% and 28.0%, respectively, and the reduced relative proportions were 18.4% and 29.7%.

Table 4.16 shows the percentage of NPs that are unmodified, modified by just an RC or prepositional phrase (PP), or modified by both an RC and a PP. Results are given for all NPs and for NPs that serve as matrix subjects, matrix direct objects, or are nested within another relative clause. The reason for breaking the results down this way is that the frequency of RC modification is expected to differ depending on the role or location of the modified noun and that this difference will have an important effect on the difficulty of processing various types of relative clauses. Because the two corpora agree quite well in regard to RC statistics, their results have been combined.

Table 4.16 contains a number of interesting results. Matrix subjects and nested NPs pattern very much like NPs as a whole. PP-modification of NPs is much more common than RC-modification. The frequency of combined RC- and PP-modification is a bit less than half of what one might expect from the product of their independent likelihoods. Matrix objects are two or three times more likely to be modified by either RCs or PPs than are other NPs. Noun-modifying PPs are discussed further in Section 4.6.

The next set of analyses examines the types of relative clauses used to modify nouns in various roles. Table 4.17 shows the overall distribution of marked and reduced RCs. PP-extracted, adverbial-extracted, and infinitival RCs have been eliminated and the remaining clauses have been classified as subject-relative, object-relative, or passive. The subject relatives have been further broken down by whether the verb in the RC is intransitive, transitive, or uses a

Extraction Type	Marked		Reduced		Total	
Subject-Intransitive	14.9%	695	5.0%	234	20.0%	929
Subject-Transitive	26.4%	1,227	7.7%	359	34.1%	1,586
Subject-Sent. Comp.	1.3%	62	0.2%	10	1.6%	72
Passive	4.5%	207	29.7%	1,381	34.1%	1,588
Object	3.0%	140	7.3%	340	10.3%	480
Total	50.1%	2,331	49.9%	2,324		4,655

Table 4.18: Distribution of selected RC types modifying matrix subjects.

Extraction Type	Marked		Reduced		Total	
Subject-Intransitive	18.2%	600	4.8%	157	23.0%	757
Subject-Transitive	21.9%	722	8.2%	270	30.1%	992
Subject-Sent. Comp.	1.7%	56	0.7%	22	2.4%	78
Passive	5.2%	172	23.1%	760	28.3%	932
Object	6.3%	208	10.0%	330	16.3%	538
Total	53.3%	1,758	46.7%	1,539		3,297

Table 4.19: Distribution of selected RC types modifying matrix objects.

sentential complement, as in, "*The grad student who thought he could write his thesis in a month was sorely mistaken*."

Tables 4.18, 4.19, and 4.20 show the distribution of RCs modifying matrix subjects, matrix objects, and nested NPs, respectively. Subject-modifying RCs are evenly split between marked and reduced relatives. Subject-extractions account for 55.7% of subject-modifying RCs, with 34.1% passive and only 10.3% object-extracted. Object-modifying RCs, in Table 4.19, are somewhat less likely to be reduced. They are more likely to have intransitive subject-extracted RCs but less likely to have transitive ones. This makes some sense if objects tend to be inanimate and thus less likely to use a transitive verb. Object-modifying RCs are also more likely to be modified by an object-relative. This too makes some intuitive sense. However, objects are actually less likely to be modified by a passive RC, which is a bit hard to explain.

RCs modifying nouns that are nested within other RCs tend to be reduced more often than those modifying matrix nouns. Nested RCs are also more likely to be passive and less likely to be transitive than other RCs.

4.3.3 Nested relative clauses

Because the comprehensibility of nested RCs, or one RC contained within another, has received so much attention in the psycholinguistic literature, it is interesting to look further into the frequency of these structures in written text and, in particular, into the relationship between the nested and the parent clauses. Ideally, it would have been nice to produce a complete analysis of nested RCs, crossing such factors as the clause type and context of both clauses.

Extraction Type	Marked		Reduced		Total	
Subject-Intransitive	13.6%	294	6.8%	146	20.3%	440
Subject-Transitive	18.9%	409	8.1%	175	27.0%	584
Subject-Sent. Comp.	0.7%	15	0.3%	6	1.0%	21
Passive	5.2%	112	31.6%	684	36.8%	796
Object	4.4%	95	10.5%	228	14.9%	323
Total	42.7%	925	57.3%	1,239		2,164

Table 4.20: Distribution of selected RC types modifying nested nouns.

Parent Clause	Nested Clause	WSJ		BC		Total	
Marked	Marked	39.1%	225	40.8%	296	40.0%	521
Marked	Reduced	60.9%	351	59.2%	430	60.0%	781
Marked	Total	7.2%	576	9.2%	726	8.2%	1,302
Reduced	Marked	43.8%	248	47.0%	282	45.5%	530
Reduced	Reduced	56.1%	318	53.0%	318	54.5%	636
Reduced	Total	6.6%	566	6.5%	600	6.6%	1,166

Table 4.21: Nested relative clauses and reduction.

However, because of the difficulty of parsing and because of the relative infrequency of nested clauses, the complete breakdown of nested RCs was not done. The analysis here simply focuses on whether the parent and embedded RCs are reduced or marked. Table 4.21 summarizes the results. To be clear, *parent clause* here refers to the clause in which the other one is contained.

Of all 7,983 marked RCs in the WSJ and the 7,862 in the BC, 7.2% and 9.2%, respectively, contained an embedded RC. Of the 8,545 reduced RCs in the WSJ and the 9,169 in the BC, 6.6% and 6.5%, respectively, contained an embedded RC. In comparison, the overall probability of a noun phrase being modified by a marked RC was 1.8% in the WSJ and 2.1% in the BC. The probability of NP modification by reduced relatives was 2.0% and 2.5%, respectively.

Several observations can be made about this data. Interestingly, noun phrases within RCs are about three times as likely as the average NP to be modified by a nested RC. One possible explanation for this may be that NPs inside of RCs tend to be ordinary, concrete NPs and thus permit modification, whereas the set of all NPs includes many non-canonical NPs, such as dollar amounts, which may not permit RC-modification. However, a cursory look indicates that non-canonical NPs, or those not headed by an `NN` noun, account for fewer than 25% of NPs. Thus, this is at best a partial explanation.

For whatever reason, the BC sources tend to use slightly more RCs and a greater proportion of reduced relatives. Across the two corpora, reduced relatives make up 52.8% of all RCs. However, the proportion of reduced relatives is greater among nested clauses (57.4%), particularly within marked parent clauses (60.0%). Thus, there appears to be a tendency to reduce nested relative clauses in written English, especially if the parent clause is marked.

Furthermore, a marked RC is more likely to contain a nesting than is a reduced RC. Similarly, given that a parent clause contains a nested relative clause, the probability that the parent clause is reduced is only 47.2%, somewhat less than the overall rate of 52.8%. There are at least two explanations for this. It is possible that reduced RCs, because they tend to be passive or object-extractions, do not permit, semantically or pragmatically, nested modification. Alternatively, it may be that a writer will be more likely *not* to reduce a parent clause knowing that the clause will contain an embedding. Possibly, both factors are at work. To tease them apart, one would need to study the frequency of nested modification in conjunction with a deeper analysis of the parent clause type.

4.3.4 Personal pronouns in relative clauses

One final issue is the use of pronouns in relative clauses. Gibson and Warren (1998) found that sentences with nested RCs are easier to comprehend if an indexical (first- or second-person) pronoun is in the subject position of the nested clause. One might ask how frequently pronouns occur in relative clauses and whether they are used more frequently in nested clauses than chance would dictate.

Table 4.22 shows the 20 pronouns that most frequently occur in the NP position within relative clauses in the two corpora. Separate counts are given for RCs overall and for nested RCs. These counts include all pronouns, not just those in subject position. Unfortunately, because the corpora contain so little conversational material, the frequency of indexical pronouns is undoubtedly much lower than in spoken language. Nevertheless, the frequency of pronouns in written English may be of interest to some researchers.

Overall, 2,006 of the 15,845 WSJ RCs (12.7%) and 2,101 of the 17,714 (11.9%) BC RCs contain pronouns. Of the 1,302 nested RCs in the WSJ and the 1,166 nested RCs in the BC, 148 (11.4%) and 336 (28.8%), respectively,

	All Relative Clauses			Nested Clauses		
Pronoun	**WSJ**	**BC**	**Total**	**WSJ**	**BC**	**Total**
it	663	436	1,099	51	75	126
he	342	480	822	36	102	138
they	366	261	627	36	69	105
him	77	266	343	12	40	52
them	166	163	329	14	37	51
we	153	151	304	7	37	44
I	107	175	282	9	47	56
you	112	142	254	7	31	38
she	68	129	197	8	31	39
her	25	90	115	3	17	20
us	31	68	99	7	12	19
me	22	68	90	1	9	10
himself	20	60	80	1	5	6
themselves	28	49	77	6	9	15
itself	16	38	54	2	9	11
one	6	12	18	0	1	1
herself	1	11	12	0	2	2
ourselves	2	7	9	0	4	4
theirs	0	5	5	0	1	1
his	1	3	4	0	0	0

Table 4.22: Pronouns appearing within relative clauses.

contain pronouns. Therefore, pronouns do not appear to be used with greater frequency in nested RCs in the WSJ, but they are in the BC. Perhaps these differences are due to the fictional content of the BC.

4.4 Sentential noun phrases

Next to relative clauses, sentential noun phrases (SNPs) may be one of the most-studied structures in psycholinguistics. An SNP is a phrase composed of a sentence with an optional introductory conjunction, such as *that the dog ran away*. Although these usually appear as a complement or object of a verb, *The boy was sad that the dog ran away*, they may also be used in subject position, *That the dog ran away saddened the boy*.

Table 4.23 shows the most common types of sentential complements (SCs) and Table 4.24 shows the most common sentential subjects. The fact that the columns add up to a bit more than the total number of phrases is a result of classification overlap. The types are as follows:

SNP: If an SC, this consists of an `SBAR`, immediately dominated by a `VP`, that is not headed by *as*, *because*, *while*, *though*, or *so*. Sentential subjects appear to be encoded in the WSJ as an `SBAR-SBJ` or `SBAR-NOM-SBJ` dominated by an `S` that does not also dominate an `ADJP-PRD`.

SNP-reduced: A reduced SNP having no complementizer.

SNP-that: An SNP introduced by *that*.

SNP-whether: An SNP introduced by *whether*.

SNP-if: An SNP introduced by *if*.

SNP-for: An SNP introduced by *for*.

W %	B %	W #	B #	Class	Description
49.5%	25.3%	6,835	1,790	SNP-reduced	Reduced SNP
27.4%	49.7%	3,790	3,518	SNP-that	*that these events took place*
1.04%	1.55%	143	110	SNP-whether	*whether the ad is truthful*
0.40%	2.22%	55	157	SNP-if	*if charges should be brought*
0.25%	0.14%	34	10	SNP-for	*for prices to rise*
0.22%	1.13%	30	80	SNP-other	A variety, often mis-tagged
1.87%	11.0%	259	781	SNP-WHNP	*what the answer was*
1.75%	10.0%	242	707	SNP-WHADVP	*how the other team does*
0.11%	0%	15	0	SNP-WHADJP	*how serious the other team is*
0.06%	0.11%	8	8	SNP-WHPP	*in which district to run*
18.0%	25.3%	2,488	0	SNP-NONE	Empty SNP

Table 4.23: Analysis of sentential complements: 13,814 from the WSJ and 7,074 from the BC.

W %	B %	W #	B #	Class	Description
5.9%	8.4%	8	23	SNP-that	*That membership hasn't fallen is...*
8.8%	5.5%	12	15	SNP-whether	*Whether we should go is...*
1.5%	0%	2	0	SNP-if	*If Wang will stage a comeback is...*
2.2%	1.1%	3	3	SNP-for	*For us to leave would be...*
76%	78%	103	213	SNP-WHNP	*What she did was..., What they did is...*
5.2%	7.6%	7	21	SNP-WHADVP	*Why he did that isn't clear.*
0.7%	0%	1	0	SNP-WHADJP	*Just how mean he can be is...*
0.7%	0.7%	1	2	SNP-WHPP	*To what extent that applies is...*

Table 4.24: Analysis of the sentential subjects: 136 from the WSJ and 275 from the BC.

SNP-other: An SNP introduced by a conjunction other than *that*, *whether*, *if*, or *for*. Many of these are due to mis-taggings.

SNP-WHNP: An SNP headed by *what*, *who*, *which*, *how much*, etc. For example, *I knew what the answer was*.

SNP-WHADVP: An SNP usually headed by *how* or *why*: *We'll see how the other team does*.

SNP-WHADJP: An SNP usually introduced by *how* followed by an adjective: *We'll see how serious the other team is*.

SNP-WHPP: An SNP headed by a prepositional phrase: *He is deciding in which district to run*.

SNP-NONE: An empty SNP. These are due to gapped or transformed constructions.

While syntax statistics in the WSJ and BC agree quite well in general, there are some major differences in the distribution of SNPs between the corpora. This is attributable to the fact that SNP use is highly dependent on the speaker's tone. News articles tend to make frequent use of SNPs, especially those used to report what someone said. There are also many more reduced SNPs in the WSJ, accounting for half of all SNPs. The BC contains more SNP-WHNPs and SNP-WHADVPs, as these phrases are more common in informal, colloquial language. It is true, however, that SNPs introduced by *that* are by far the most common marked form in either corpus.

Sentential subjects, shown in Table 4.24, are substantially less common than SCs, by several orders of magnitude. Sentential subjects are more frequent in the BC than in the WSJ, although the opposite is true of SCs. Most sentential subjects are of the SNP-WHNP form. Relatively few are of the more "classic" SNP-that or SNP-whether varieties, which have received some attention in the empirical literature (Gibson & Thomas, 1995). Sentential subjects seem to be employed in either overly sophisticated (*That such expansion can be obtained without a raise in taxes is due to...*) or overly colloquial (*What he did was...*) language, but are seldom used in good writing.

Table 4.25 contains the 25 verbs that are most frequently associated with an SC. The percentages under the heading "With Sentential Complement" indicate the probability that the verb is followed by an SC. The percentages under the heading "With Reduced SC" indicate the proportion of the SCs following the verb that are reduced.

	Overall			With Sentential Complement						With Reduced SC					
Verb	WSJ	BC	Total	WSJ		BC		Total		WSJ		BC		Total	
said	7,133	1,954	9,087	80%	5,741	18%	367	67%	6,108	68%	3,949	68%	251	68%	4,200
says	2,467	200	2,667	48%	1,190	34%	68	47%	1,258	45%	546	61%	42	46%	588
say	860	493	1,353	84%	729	37%	187	67%	916	59%	432	36%	69	54%	501
think	389	431	820	80%	313	56%	242	67%	555	87%	274	84%	204	86%	478
know	253	682	935	50%	128	43%	299	45%	427	32%	41	22%	66	25%	107
is	8,446	7,510	15,956	0.3%	23	4%	323	2%	346	4%	1	2%	7	2%	8
told	226	412	638	65%	148	39%	163	48%	311	30%	45	39%	65	35%	110
believe	201	200	401	89%	179	57%	115	73%	294	65%	117	41%	48	56%	165
thought	120	413	533	62%	75	35%	148	41%	223	85%	64	86%	128	86%	192
knew	29	395	424	62%	18	51%	202	51%	220	66%	12	39%	79	41%	91
see	355	769	1,124	15%	56	18%	145	17%	201	5%	3	5%	8	5%	11
added	379	151	530	45%	172	11%	17	35%	189	6%	11	11%	2	6%	13
saying	166	109	275	78%	131	44%	48	65%	179	74%	97	35%	17	63%	114
was	4,878	6,306	11,184	0.7%	32	2%	131	1%	163	0%	0	0%	0	0%	0
noted	173	81	254	79%	138	25%	21	62%	159	5%	7	0%	0	4%	7
reported	473	118	591	25%	121	22%	26	24%	147	23%	28	7%	2	20%	30
tell	87	266	353	56%	49	36%	96	41%	145	34%	17	25%	24	28%	41
found	220	534	754	33%	73	11%	64	18%	137	15%	11	28%	18	21%	29
show	151	204	355	36%	55	38%	78	37%	133	20%	11	5%	4	11%	15
believes	109	43	152	90%	99	48%	21	78%	120	70%	70	23%	5	62%	75
asked	200	398	598	18%	37	17%	71	18%	108	0%	0	0%	0	0%	0
announced	261	88	349	29%	77	30%	27	29%	104	37%	29	18%	5	32%	34
indicated	78	107	185	80%	63	32%	35	52%	98	44%	28	20%	7	35%	35
felt	63	352	415	33%	21	21%	76	23%	97	76%	16	40%	31	48%	47
means	106	106	212	52%	56	32%	34	42%	90	71%	40	26%	9	54%	49

Table 4.25: Verb forms frequently taking a sentential complement.

SCs most often follow the verbs of saying, especially in the WSJ. In the BC, but not in the WSJ, they sometimes follow *is* and *was*. These cases appear mainly in informal expressions such as, *The thing is that…* which are rare in news articles. *Believe* and *believes* are the two verbs most likely to take an SC, followed by *say* and its variants.

Juliano and Tanenhaus (1993) argued that higher frequency verbs are more likely to take reduced SCs. This finding is not entirely supported by the WSJ and BC data. Of the top 50 verbs most frequently occurring with a sentential complement, the correlation between overall verb frequency and the proportion of reduced SCs was -0.14. However, much of this null effect is due to closed-class verbs like *is* which only rarely take SCs and almost never reduced ones. If we ignore the verbs *is*, *'s*, *was*, and *be*, the correlation between frequency and reduction increases to 0.23.

But there are stronger predictors of SC reduction. The correlation between number of occurrences of the verb with an SC and the proportion of reduced SCs was 0.26. Better yet is the probability the verb is followed by an SC, given that the verb occurs, which has a correlation of 0.42. We might therefore conclude that it is not the overall verb frequency that leads speakers to prefer reduced SCs, but the likelihood that an SC will occur. Essentially, if an SC is expected, it can safely be reduced without causing confusion.

It is interesting to note that some verbs rarely or never take a reduced SC. The verb *asked* is a good example. Unlike many verbs that use SCs, *asked* is often followed by a noun phrase, indicating the person to whom the question is addressed. A marked SC in this case leads to a strong garden path and is generally avoided. *See*, *show*, *find*, and *adds* similarly take direct objects and rarely use a reduced SC. However, some other verbs, including *determine*, *argue*, and *noted*, do not usually take a direct object but still strongly prefer marked SCs.

	All NPs		Matrix Subject				Matrix Object			
Article			**Unmodified**		**Modified**		**Unmodified**		**Modified**	
	Singular									
the	41.70%	109,664	46.90%	15,118	50.90%	5,345	21.50%	3,273	28.50%	3,096
-none-	40.80%	107,283	35.90%	11,552	26.90%	2,822	45.30%	6,881	24.10%	2,619
a/an	22.00%	57,890	8.47%	2,727	18.70%	1,966	26.40%	4,019	42.50%	4,608
this	2.52%	6,609	2.67%	859	1.72%	181	2.28%	346	0.65%	71
that	0.85%	2,228	1.06%	343	0.44%	46	0.70%	106	0.24%	26
no	0.78%	2,040	0.82%	265	0.59%	62	2.16%	328	2.76%	299
any	0.75%	1,970	0.22%	71	0.71%	75	0.43%	66	0.55%	60
some	0.48%	1,255	0.12%	39	0.22%	23	0.66%	101	0.68%	74
each	0.39%	1,037	0.57%	182	0.32%	34	0.28%	42	0.00%	0
another	0.38%	1,001	0.48%	154	0.58%	61	0.51%	78	0.72%	78
all	0.31%	803	0.13%	42	0.10%	10	0.18%	27	0.00%	0
every	0.27%	718	0.23%	73	0.66%	69	0.28%	43	0.16%	17
	Plural									
-none-	75.30%	92,649	71.70%	13,938	68.1%	3,477	79.90%	4,626	75.10%	2,415
the	23.90%	29,358	20.90%	4,065	27.2%	1,390	13.80%	800	19.20%	619
these	1.64%	2,018	2.85%	555	0.96%	49	1.38%	80	0.25%	8
a few	1.44%	1,771	0.42%	82	0.49%	25	2.50%	145	1.65%	53
some	1.35%	1,662	3.31%	643	1.63%	83	1.59%	92	1.68%	54
all	1.05%	1,298	0.70%	137	1.16%	59	0.62%	36	0.75%	24
those	0.63%	779	0.77%	149	0.55%	28	0.54%	31	0.40%	13
both	0.42%	517	0.74%	143	0.24%	12	0.36%	21	0.09%	3
no	0.39%	474	0.35%	69	0.25%	13	1.23%	71	0.90%	29
any	0.37%	459	0.12%	24	0.31%	16	0.55%	32	0.40%	13

Table 4.26: Common determiners as a function of position, number, and post-modification.

4.5 Determiners and adjectives

Increasing attention has been given recently to the effect of determiners and adjectives in influencing the comprehension and reading times of object- and subject-extracted RCs, ambiguous PP attachments, and the main verb/reduced relative ambiguity (Ni & Crain, 1990; Sedivy & Spivey-Knowlton, 1994). The word *that* has also been of interest because of its many uses, including its role as a complementizer (*Dave knew that he was late.*), a relative pronoun (*The thing that Dave knew…*), a demonstrative pronoun (*Dave knew that.*), and as a determiner (*Dave knew that answer.*) (Juliano & Tanenhaus, 1993). It is commonly argued that post-modification of an indefinite NP is more felicitous than post-modification of a definite NP (introduced by *the*, *this*, or *that*). It has also been suggested that modification by an adjective should bias readers against further post-modification by an RC (Ni & Crain, 1990) or a PP (Britt, 1994). We therefore investigate the frequency of determiners and adjective modification in conjunction with possible post-modification.

4.5.1 Articles and quantifiers

Table 4.26 shows the frequency of occurrence of the most common articles and quantifiers across the two corpora as a function of whether the noun is singular or plural and whether the noun is unmodified or is post-modified by either a PP or RC. Proper nouns are not included in the counts. Singular nouns with no determiner are generally mass nouns, with the exception of a few special words like *yesterday*. Summary data are shown in Table 4.27 where definite articles (*the*, *this*, *these*, *that*, and *those*) are isolated from indefinite articles and quantifiers.

Overall, definite articles are used about twice as often as indefinites. Singular matrix subjects tend to be definite. Singular matrix objects, however, are more likely to be indefinite. As expected, the percentage of indefinites is higher for modified singular nouns than for unmodified ones. However, the percentage of definites is also a bit higher with modification, which is attributable to the fact that the percentage of no determiners is much lower.

Article	All NPs		Matrix Subject Unmodified		Matrix Subject Modified		Matrix Object Unmodified		Matrix Object Modified	
	Singular									
-none-	36.7%	107,283	36.8%	11,552	26.4%	2,822	44.9%	6,881	23.9%	2,619
Definite	40.5%	118,501	51.9%	16,320	52.1%	5,572	24.3%	3,725	29.2%	3,193
Indefinite	22.8%	57,890	11.3%	3,553	21.5%	2,300	30.7%	4,704	46.9%	5,136
	Plural									
-none-	67.5%	92,649	70.4%	13,938	67.5%	3,477	78.0%	4,626	74.7%	2,415
Definite	23.4%	32,155	24.1%	4,769	28.5%	1,467	15.4%	911	19.8%	640
Indefinite	9.0%	12,362	5.5%	1,098	4.0%	208	6.7%	397	5.5%	176

Table 4.27: The data from the previous table classified by definiteness.

Adjective?	All NPs		Matrix Subject Unmodified		Matrix Subject Modified		Matrix Object Unmodified		Matrix Object Modified	
No adjective	82.0%	660,985	90.0%	107,596	76.5%	14,630	76.7%	21,097	67.5%	10,564
Adjective	18.0%	144,765	10.0%	11,923	23.5%	4,483	23.3%	6,418	32.5%	5,081

Table 4.28: Frequency of adjective modification as a function of position and post-modification.

Most plural nouns usually have no determiner and they are rarely indefinite. Singulars are most often definite, but also frequently have no determiner. Matrix subjects are more likely than matrix objects to be definite and less likely to be indefinite. Interestingly, with modification the percentage of definite plurals increases and the percentage of indefinite plurals decreases, which is unexpected.

4.5.2 Adjectives

Along with determiners, adjectives are also thought to be related to the likelihood that a noun is post-modified by a PP or RC. It is generally held, with some empirical justification, that nouns modified by adjectives should be less likely to have further modification. In fact, that is not the case in the WSJ and Brown corpora.

Table 4.28 gives the proportion of adjective modification as a function of noun role and post-modification. One main effect in this data is that adjectives are more common with matrix objects than they are with matrix subjects. This may not be too surprising. But, unexpectedly, post-modified nouns are *more* likely to have adjectives than are unmodified nouns. Conversely, nouns with adjectives are more likely to be post-modified than nouns without adjectives.

This surprising result may have a reasonable explanation. Post-modified nouns seem to have a high percentage of quantitative adjectives, such as *only*, *first*, *most*, and *other*, and comparative adjectives, like *highest*. Although some may consider quantifiers to be determiners, the Treebank tags most of them as adjectives. It is relatively rare for post-modified nouns to have descriptive adjectives, but the aforementioned studies that found effects of adjectives on the processing of further modification used descriptive adjectives (Ni & Crain, 1990; Britt, 1994). It should be noted, however, that Ni and Crain (1990) contrasted the descriptive adjectives with a condition using *only*, which did lead subjects to be faster at reading an ambiguous post-modifier, and that Britt (1994) preceded the target sentences with a context that made the adjective modified noun an unambiguous reference.

It seems that the present analysis of adjectives and post-modification is not sufficient to fully understand their interaction. A better analysis must discriminate among different forms of adjectives, including, but perhaps not limited to, quantifying, comparative, and descriptive adjectives. Empirical researchers should be careful not to treat all adjectives the same. Very different results might be expected with descriptive adjectives and quantifiers.

W %	B %	W #	B #	Class	Description
47.4%	42.8%	50,737	52,112	NP-PP	Noun-modifying
37.6%	35.0%	40,249	42,601	VP-PP	Verb-modifying
7.11%	11.6%	7,607	14,097	S-PP	Shifted away from point of attachment
2.66%	3.33%	2,843	4,053	PP-PP	Compound PPs
2.68%	4.11%	2,864	5,007	ADJP-PP	Adjective-modifying
1.49%	1.83%	1,592	2,228	ADVP-PP	Adverb-modifying
0.32%		344		PRN-PP	Parenthetical, *", for example,"*
0.17%		186		NAC-PP	*Bank of New York*

Table 4.29: Sites of prepositional phrase attachment.

4.6 Prepositional phrases

Prepositional phrases (PPs) have received the most attention recently in connection with attachment ambiguities. The following sentences illustrate the complexity of PP attachment in English:

(42a) The man saw the bird with binoculars.

(42b) The bird saw the man with binoculars.

In the first case, *with binoculars* is typically interpreted as modifying the verb *saw*, while in the second sentence it modifies *the man*, despite similar word order.

Unfortunately, detecting potentially ambiguous attachments would be very difficult or impossible in these corpora as currently tagged. Nevertheless, a preliminary analysis of prepositional phrases was conducted. There were a total of 107,055 PPs in the WSJ and 121,722 in the BC. These are broken down by site of attachment in Table 4.29.

The labels are as follows:

NP-PP: A PP modifying a noun-phrase.

VP-PP: A PP modifying a verb-phrase.

S-PP: A PP dominated by an S. Often these are noun- or verb- modifying but have been detached from the site of modification. One must be careful in extracting these from the BC as many subordinate clauses, such as, *Until the boy ran away*, are incorrectly labeled PPs.

PP-PP: A PP dominated by another PP. Often these involve the constructions *according to...* or *from X to Y*, or phrases like *to 30% in May*. They are also used in coordinate PPs.

ADJP-PP: A PP modifying an adjective, as in *compatible with international standards*.

ADVP-PP: A PP modifying an adverb, as in *away from home*.

PRN-PP: Used for parenthetical remarks, such as *", for example,"* or *", after all,"*. Not used in the BC.

NAC-PP: Proper names or titles involving PPs, such as *Secretary of State* or *Americans with Disabilities*. These are not used in the BC. They are used only sporadically in the WSJ. For example, *Bank of New York* occurs 22 times in the WSJ but in only 3 of these occasions was it encoded using a NAC-PP.

Overall, the number of noun-modifying PPs was somewhat greater than the number of verb-modifying PPs. However, the likelihood that a given NP would be modified (11.7% WSJ, 14.0% BC) was about half the modification rate of VPs (22.3% WSJ, 28.9% BC).

Table 4.30 shows the 25 prepositions most frequently used to modify nouns, one the left, and verbs, on the right. Some prepositions, such as *with*, can take a variety of meanings: instrument, accompaniment, manner, etc. With the newer tagging style used in the WSJ, which distinguishes between different classes of PP, it ought to be possible to study these separate uses. However, over 65% of *with* PPs in the WSJ were tagged using a non-specific PP designation. Therefore, it may not be possible to automatically compile an accurate count of preposition uses.

Preposition	WSJ	BC	Total
of	25,809	32,244	58,053
in	7,960	5,697	13,657
for	4,497	3,058	7,555
to	1,894	2,183	4,077
on	2,342	1,720	4,062
with	1,640	1,549	3,189
at	1,556	1,154	2,710
from	1,512	1,109	2,621
as	499	902	1,401
by	795	595	1,390
about	465	414	879
between	347	452	799
than	257	388	645
over	351	200	551
against	277	179	456
like	170	276	446
such	395	9	404
per	93	292	385
under	120	128	248
after	105	132	237
among	140	92	232
into	103	120	223
before	74	122	196
toward	70	118	188
via	168	1	169

Preposition	WSJ	BC	Total
in	8,418	8,343	16,761
to	6,469	6,010	12,479
by	4,336	3,729	8,065
for	4,149	3,533	7,682
with	2,614	4,150	6,764
on	3,233	3,149	6,382
at	2,984	2,468	5,452
from	3,089	2,304	5,393
as	1,486	1,849	3,335
into	958	1,518	2,476
of	588	1,282	1,870
about	446	702	1,148
through	510	602	1,112
over	403	532	935
out	235	548	783
under	420	336	756
like	197	536	733
against	282	356	638
after	473	154	627
without	238	301	539
during	359	109	468
before	245	207	452
up	50	309	359
upon	37	316	353
until	158	190	348

Table 4.30: Most common noun-modifying (left) and verb-modifying (right) prepositions.

Conjunction	WSJ		BC		Total	
and	84.6%	12,334	82.1%	12,689	83.3%	25,023
or	13.6%	1,980	15.3%	2,358	14.5%	4,338
but	0.64%	93	1.54%	238	1.10%	331
plus	0.26%	38	0.26%	40	0.26%	78
nor	0.26%	38	0.35%	54	0.31%	92
both	0.19%	28	0.01%	2	0.10%	30
neither	0.15%	22	0.19%	30	0.17%	52
either	0.14%	21	0.16%	25	0.15%	46

Table 4.31: Common noun phrase conjunctions.

Conjunction	WSJ		BC		Total	
and	79.7%	3,830	88.1%	5,438	84.4%	9,268
or	9.20%	442	8.03%	496	8.54%	938
but	10.3%	496	3.35%	207	6.40%	703
either	0.35%	17	0.08%	5	0.20%	22
nor	0.19%	9	0.31%	19	0.26%	28

Table 4.32: Common verb phrase conjunctions.

4.7 Coordination and subordination

4.7.1 Coordinate phrases and clauses

Coordination, or the combining of phrases or clauses having roughly equal importance, is often cited as giving language its infinite syntactic productivity. Whether or not one accepts the grammaticality of phrases composed by extended use of conjunction, coordination is still an important aspect of any language.

Coordination was studied in the WSJ and BC for four constructions: NPs, VPs, adjective phrases, and clauses. Overall, 3.7% of noun phrases contained coordinating constructions, as did 3.3% of VPs, 7.3% of adjectives, and 5.6% of clauses. The conjunctions most commonly used in these constructions are shown in Tables 4.31 through 4.34.

Among the phrases, *and* is by far the most common conjunction, with *or* running a distant second. Among clauses, *but* was a close second to *and*. Interestingly, *but* was more frequent than *and* in the WSJ, but much less frequent in the BC.

4.7.2 Subordinate clauses

The final topic addressed in this study is that of subordinate clauses. Detecting subordinate clauses in these corpora is actually a bit tricky. In both corpora they are encoded as an `SBAR*` dominated by an `S`. However, in the WSJ they are also labeled with an `SBAR-ADV`, `SBAR-TMP`, `SBAR-PRD`, or `SBAR-LOC` dominated by a `VP`. In the BC, subordinate

Conjunction	WSJ		BC		Total	
and	79.6%	850	79.4%	1,969	79.5%	2,819
or	12.1%	129	13.7%	339	13.2%	468
but	7.21%	77	5.52%	137	6.03%	214
nor	0.47%	5	0.85%	21	0.73%	26

Table 4.33: Common adjective conjunctions.

Conjunction	WSJ		BC		Total	
and	45.5%	2,779	63.0%	5,722	56.0%	8,501
but	52.4%	3,196	33.0%	2,996	40.8%	6,192
or	1.07%	65	2.59%	235	1.98%	300
yet	0.51%	31	0.37%	34	0.43%	65
nor	0.23%	14	0.43%	39	0.35%	53
so	0.20%	12	0.45%	41	0.35%	53
either	0.07%	4	0.07%	6	0.07%	10

Table 4.34: Common coordinating clause conjunctions.

Conjunction	WSJ		BC		Total	
when	13.6%	1,028	16.3%	1,606	15.2%	2,634
if	15.2%	1,146	15.0%	1,473	15.1%	2,619
as	12.9%	971	11.2%	1,102	11.9%	2,073
while	10.6%	798	5.15%	506	7.50%	1,304
because	10.2%	768	4.90%	482	7.19%	1,250
that	3.71%	280	8.30%	816	6.30%	1,096
although	4.24%	320	2.84%	279	3.45%	599
after	3.77%	285	2.64%	260	3.13%	545
since	2.10%	159	3.56%	350	2.93%	509
before	2.28%	172	3.22%	317	2.81%	489
even	3.02%	228	1.73%	170	2.29%	398
though	2.14%	162	1.82%	179	1.96%	341
in	0.97%	73	2.60%	256	1.89%	329
until	2.02%	153	1.70%	167	1.84%	320
so	1.16%	88	2.04%	201	1.66%	289
for	0.46%	35	1.64%	161	1.13%	196
unless	1.28%	97	0.83%	82	1.03%	179
with	0.90%	68	1.04%	102	0.98%	170
which	0.61%	46	1.20%	118	0.94%	164
where	0.62%	47	1.10%	108	0.89%	155
once	1.14%	86	0.52%	51	0.79%	137
just	0.75%	57	0.68%	67	0.71%	124
what	0.52%	39	0.69%	68	0.62%	107
whether	0.48%	36	0.57%	56	0.53%	92
how	0.29%	22	0.53%	52	0.43%	74

Table 4.35: Common subordinating clause conjunctions.

clauses are often coded as prepositional phrases, but are distinguishable from true PPs because they dominate an `S`.

Subordinating conjunctions are roughly as common as coordinating conjunctions overall, but they are much more diverse. Table 4.35 shows the 25 most common subordinating conjunctions in the corpora. Some of these may be incorrectly classified as subordinating conjunctions. For example, clauses headed by *that* are most likely sentential noun phrases that are not enclosed within an `NP`.

4.8 Conclusion

This chapter has presented an analysis of the distribution of selected syntactic structures in the Wall Street Journal and Brown corpora, the immediate goal of which was to discover statistics that will be used in constructing the training language for the CSCP model. While it is hoped that the results are interesting and useful to some readers in their present form, most researchers will have different goals and may want to conduct their own analyses or expand on the present ones. Such readers may benefit from adopting the methods used in conducting this study. For corpus analysis in general, TGREP2 should be a substantial improvement over TGREP. Using this in conjunction with the technique of partitioning sets of phrases into non-overlapping subsets enables one to quickly detect mistakes without the need for exhaustive search. Most of the investigator's time can then be spent revising search patterns, rather than filtering through vast sets of phrases.

While the Penn Treebank is a tremendously valuable resource and the result of an impressive effort, it is not without its flaws. The relative ease with which potential tagging errors are discovered when using the partitioning method suggests that developers of tagged corpora could benefit from using similar methods of analysis to help reduce mistakes. One might even take this a step further and build an explicit context-free grammar of the language. While this does not appear to have been done by the authors of the Treebank, and may be quite difficult, it could substantially improve the consistency and utility of the corpus, as well as facilitating the syntactic tagging of the corpus.

A strict grammar would establish the legal productions of each term in the parse tree. Checking for violations in this grammar would identify many parsing mistakes, such as the grouping errors that appear in complex, conjoined verb phrases and cases of present participles being marked as nouns or adjectives. Ultimately, we will have not just a consistently tagged corpus but a complete grammar of the language with associated statistics, which would be of great benefit to future endeavors of this kind.

Specifying a grammar would also force the tagging system to become more explicit and thus more helpful to users of the corpus. For example, a standard context-free grammar produces parse trees in which any two subtrees with the same root node may be exchanged without violating the grammaticality of the tree. Any subtree marked `X` ought to be grammatically interchangeable with any other subtree marked `X`. However, this constraint is not observed in many linguistic theories, which often try to minimize the number of basic elements. The Penn Treebank uses `VP` to refer to a variety of verb-related phrases, not all of which are interchangeable. Verbals, infinitives and participles are marked `VP`, but could not form a complete predicate, as most VPs can. These violations of composability produce ambiguities that make analyzing the corpus difficult.

An approach to tagging that is more conducive to automating both the tagging and subsequent analysis, is to create as many distinctions as can be consistently applied, while maintaining a hierarchy of symbol names that preserve the basic types. Indeed, the newer Treebank style has made some steps in this direction, but the principle should be applied more thoroughly. A VP that contains a conjunction of two other VPs might be encoded explicitly as `VP-CONJ`, not just `VP`, and a phrase headed by a present participle as `VP-VBG`. It is easy to abstract across such distinctions later, but more difficult to draw them from a corpus tagged in a more forgiving style.

Finally, using an explicit grammar could greatly increase the speed with which trained humans can parse corpora. Given a grammar, a computer program could take a sentence or phrase, produce the legal parsings (with varying degree of sophistication) and present the user, graphically, with a set of alternatives along with a measure of the confidence that each is the correct parse. Most of the work of the human expert would be in selecting the correct option or expanding the grammar where appropriate. This would greatly accelerate the process of tagging and reduce much of its variability. While the Penn Treebank is invaluable, the needs of computational- and psycholinguists will soon demand the annotation of much larger and more varied corpora, which will require more powerful parsing methods.

Chapter 5

The Penglish Language

This chapter describes the language used to train and test the CSCP model. Because the behavior of the model is potentially very sensitive to the environment to which it is exposed, the model can only be properly evaluated with a full understanding of the training language. Of particular importance are three points: the frequency of occurrence of structures in the language and how these vary with different lexical items, aspects of English that are not included in the language but that may be relevant to the tasks under investigation, and the way in which the semantics of the language are encoded. Since it is rather inconvenient to refer to this language without a name, it will be called *Penglish*, which is short for pseudo-English, or perhaps proto-English. Take your pick.

5.1 Language features

Chapter 3 reviewed a number of important areas of interest in psycholinguistics and identified a broad set of behaviors we expect the CSCP model to replicate. In order to run the appropriate experiments on the model, its training language must include all relevant constructions, with associated statistical and semantic constraints. Previous connectionist models of comprehension, prediction, and production have mainly used languages that were limited to simple sentences, with the possible addition of relative clauses and prepositional phrases. Most such languages avoided even simple things like articles, let alone sentential complements or subordinate clauses. But in order for the current model to perform the intended experiments, its language must be significantly more complex.

The following is a partial list of structures included in or features of the Penglish language. Some are actually required to perform certain experiments. Others are necessary because they are indirectly related to factors that are believed to affect performance on those experiments. For example, multiple verb tenses are necessary so that *was* is not a unique identifier of a passive construction, but could signal a progressive past tense, which will likely affect reading times on the following verb. Even aspects of a language that are not thought to be directly relevant to the processing of a particular type of structure may still have an indirect influence on it.

- Various verb tenses and aspects, including simple past (*ran*), past progressive (*was/were running*), past perfect (*had run*), present (*run/runs*), present progressive (*is/are/am running*), present perfect (*has/have run*), and simple future (*will run*).
- Passive voice in a variety of the tenses: *was/were asked*, *had been asked*, *is/are/am asked*, *has/have been asked*, *will be asked*.
- 56 verb stems are included. These vary in terms of argument structures, including obligatory intransitives, obligatory transitives, optional intransitives, ditransitives, verbs that take sentential complements, and verbs that take sentential complements along with a noun complement. The verbs differ in terms of such factors as their frequency of occurrence in reduced relatives and their preference for NP complements versus sentential complements.
- Sentential complements, which can be reduced or introduced by *that* or *if*.
- Several types of relative clauses, including intransitive subject-extracted (*that was eating*), transitive subject-extracted (*that bit the dog*), object-extracted (*that the dog bit*), and passive (*that was bitten [by the dog]*). RCs

can either be introduced by *that*, *which*, or *who*, or reduced. The frequency of relative clauses must be sensitive to the location of the modified noun, its article, and whether it is pre-modified by an adjective.

- Noun and verb modifying prepositional phrases (PPs). These include prepositions that can only modify nouns, others that only modify verbs, and some that can modify either nouns or verbs. The frequency with which different prepositions are used to modify a particular noun or verb is lexically sensitive, as is the NP that can occur as an object of the preposition.
- Adverbs. In addition to PPs, manner can be expressed using a post-verbal adverb.
- Coordinate clauses, which can be conjoined with *and* or *but*.
- Subordinate clauses. A subordinate clause can appear before or after the main clause and is introduced by one of four conjunctions: *if*, *because*, *while*, or *although*.
- 45 noun stems are supported, which includes 37 with distinct singular/plural forms (*cat/cats*), one with identical singular/plural forms (*fish*), two mass nouns (*food, evidence*), three obligatory singulars (*something, school, baseball*[1]), one obligatory plural (*binoculars*), and two indexical pronouns (*I, you*).
- Determiners, including *the*, *a*, *this/these*, *that/those*, and *some*.
- Adjectives, some of which have dual meanings, like *hard* (difficult or not soft) and *nice* (kind or beautiful), that depend on the modified noun.
- Several types of lexical ambiguity, including verbs and adjectives with multiple senses, and some nouns and verbs that share surface forms. Furthermore, *had* can serve as a main or auxiliary verb and *that* can play a variety of roles, including determiner, complementizer, and relative pronoun.

5.1.1 Example sentences

Designing a productive grammar for a natural language that obeys both syntactic and semantic constraints is a very difficult problem. Most grammars that appear in the literature, even so-called generative grammars, are really designed to parse valid sentences. If run in reverse, such grammars will tend to produce all sorts of nonsense that may be syntactically valid according to the relevant theory, but which violates pragmatic and semantic norms. Take, for example, the verb form *consists of*. A given noun can only plausibly be said to consist of a fairly small variety of things. A book can consist of stories or pages and dinner can consist of rice and beans. But a book rarely consists of beans, nor dinner of pages. Likewise, some verbs can only use particular argument structures with certain subjects or objects. For example, a boy or a dog can walk and a boy can walk a dog, but a dog can't walk a boy. Well, not unless it's a really big dog.

If the semantic constraints were removed from the Penglish grammar, it would produce sentences like the following:

(43a) I have had school of a manager quickly.

(43b) Boys were found to the book fiercely.

(43c) The answer gave pictures in some tests.

(43d) A cop ate in a new reporter.

While these sentences may not technically contain syntactic violations, their semantic anomalies are so overwhelming that they sound ungrammatical. A goal of Penglish was for it to produce only sentences that are reasonably valid in English. Although pains were taken to implement constraints that would avoid most semantic violations, Penglish sentences do not always sound entirely natural and frequently contain non sequiturs. Sentences (44a) through (44r) are the first 18 sentences in the model's Penglish testing set. (44s) and (44t) were selected by hand from later in the set. The test of reasonableness you should impose on these sentences is whether it is possible to conceive of a context in which a normal speaker of English might say that. For example, sentence (44j) sounds rather awkward, but it could be a reasonable response to the question, "Who is writing to a boy on my behalf?"

[1]The grammar actually includes two forms of the words *school* and *baseball*. One is the concrete form, as in, "*I saw the school*," or, "*Where's my baseball?*" The other forms refer to the concepts of school and the game of baseball in general, as in "*I like baseball*." These are obligatorily singular, but do not take articles or adjectives.

(44a) We had played a trumpet for you.

(44b) Players tell something for a father of a young boy.

(44c) The father uses old tables something gave me in the house.

(44d) A answer involves a nice school.

(44e) The new teacher gave me a new book of baseball.

(44f) Houses have had something the mother has forgotten.

(44g) Birds find you felt that lawyers put the big car in the school quickly.

(44h) The mother sees you.

(44i) A small mother that has bought old food bought that evidence.

(44j) A boy is written by the nice girl for you.

(44k) Schools will have books on the floor.

(44l) Planes are left.

(44m)Lawyers give the father old pictures of me.

(44n) The lawyers put something on a table on a floor.

(44o) The owner asked the boy if the lawyer left some birds yesterday.

(44p) You have shown that the plane for the lawyer had been had in school yesterday.

(44q) The question on a table involves the mother.

(44r) I know you.

(44s) Because the boy has had the big book in a school I am taking a table to a zoo quickly.

(44t) Reporters say the mother took me and the mother had been killed.

There are some interesting features to note in these sentences. Passives occur in (44j), (44p) and (44t). A marked relative clause occurs in (44i) and reduced object relatives appear in (44c) and (44f). (44g) contains a marked sentential complement embedded within a reduced one, and (44o) and (44p) also contain sentential complements. (44s) contains a subordinating conjunction and (44t) contains a coordinating one. Finally, (44n) has a temporary PP attachment ambiguity.

Several questionable phrases do appear in these examples. The article *a*, rather than *an* is used with the noun *answer*. The *a/an* distinction is a bit tricky to implement. *An* is used whenever the immediately following word begins with a vowel sound. But that next word is not always the head of the noun phrase, it could be an adjective. Thus whether the head noun determines the *a/an* choice depends on whether an adjective intervenes, which is difficult to encode in a context-free grammar. One could argue that the choice of *a* versus *an* is not actually a feature of the syntax of English but of later phonological processes. Because the *a/an* distinction does not seem to be relevant to any of the phenomena of interest, that complexity was left out of the Penglish grammar.

The phrase "*question on a table*" in (44q) sounds rather strange. That is because the grammar allows questions to be on tables to account for the common idiomatic phrase "*the question on the table*." But when the table is indefinite, the phrase doesn't quite work. So much for the productivity of language.

5.1.2 Missing features

Although the Penglish language includes many significant complexities of English, it is important to acknowledge those features it is missing. The following is a list, albeit incomplete, of some important limitations of Penglish relative to English:

- All sentences are independent. Messages are restricted to a single sentence and there is no sense of discourse or context.

- There are no pronouns, other than indexical ones, and no other forms of anaphora. Each noun phrase refers to a unique referent. In theory, it would be possible to introduce coreference that is confined to a single sentence, as in "While he slept, Timmy dreamt of one day being a psycholinguist."
- The language contains only statements. There are no questions or commands. This is particularly a problem for modeling development, as the majority of utterences to children are questions.[2] Furthermore, questions introduce interesting issues of gapping and movement.
- There are no predicate adjectives or nominatives. This would actually be a relatively simple addition and earlier versions of the language did have predicate adjectives, but they were removed because they were not directly relevant to any of the phenomena of interest.
- There are no modal verbs. Again this would be a relatively easy addition.
- There is no subjunctive mood, nor perfect progressive, nor non-simple future tenses.
- Although there are sentential complements modifying verbs, there are no noun-modifying sentential complements, nor sentential subjects. The inclusion of sentential subjects would allow us to address the interesting finding of Juliano and Tanenhaus (1993) involving sentence initial that-phrases, but they are fairly rare in English.
- Embeddings in Penglish are limited to a depth of two or three. This prevents us from using the model to address experiments involving deep recursion.
- There are no prepositional- or adverbial-phrase-extracted relative clauses, as in "*the city that the story takes place in...*"
- There are no pre-verb adverbs; they all follow their verbs.
- Although coordinate clauses and compound verb phrases are allowed, compound nouns and adjectives, noun lists, and multiple adjectives modifying a single noun are not included.
- There are no comparative or superlative adjectives.
- There are no appositives (*The Chihuahua, a noble breed,...*).
- There are no gerunds (*running*), gerund phrases, infinitives (*to run*), or infinitival phrases.
- There are no proper nouns, other than, arguably, the indexical pronouns.
- Although, as we'll see, the input and output word representations have phonological features, the language contains no prosody. Including this should not be too problematic, as the model is expected to be sensitive to any helpful prosodic cues that are provided, particularly those indicating clause boundaries that would otherwise be ambiguous.

5.2 Penglish grammar

There are two main approaches that one might consider in generating sentences in an artificial language, where we must produce both the surface form of the sentence and its semantic representation, or meaning. One possibility is to first produce the meaning of the sentence by selecting and combining propositions. The meaning could then be translated into a sentence in the language. Another approach is to produce the sentence using a syntactic grammar and to then derive the appropriate meaning from the sentence.

The first approach may be intuitively more natural and seems to better reflect the processes by which humans produce language. However, this method would create problems for a study of this kind. Many of the statistics that must be controlled in order to perform the intended experiments are syntactic in nature. It is much easier to translate frequencies derived from corpus analyses into syntactic constraints than it is to translate them into constraints on semantic propositions. Therefore, the second, syntax-first, approach was used to generate Penglish sentences.

The grammar for the Penglish language is built around the framework of a stochastic context-free grammar (SCFG). Context-free grammars are fairly convenient for representing the syntax of most natural languages. SCFGs are also a good choice because they are quite easy to work with. Given an SCFG, it is quite easy to generate sentences, parse sentences, and perform optimal prediction, which involves producing the frequency distribution of possible next words following any sentence fragment.

[2] As determined by an analysis of child-directed speech in the CHILDES database (MacWhinney, 1999).

The problem with using SCFGs to produce a language is that it is rather difficult to implement the semantic and agreement constraints of natural language in a context-free grammar. Such constraints essentially introduce context-sensitivity. If subject/verb agreement is to be enforced, a context free grammar cannot simply have a rule that generates a subject NP followed by a predicate. It must generate a different pairing of NP and predicate for every legal combination of noun and verb, including distinctions between singulars and plurals. Such a complex grammar could not be practically written by hand.

A solution is provided by a program I have written called the Simple Language Generator, or SLG. SLG allows the user to specify the framework of a grammar in context-free form. However, additional constraints can then be placed upon the grammar in the form of constraint functions. For example, one function might constrain the choice of subject given its verb. This function could be responsible for enforcing both semantic and agreement constraints. Wherever there is a subject/verb relationship in the grammar, the subject/verb constraint function is applied, causing the choice of verbs to constrain the choice of nouns.

SLG takes the SCFG and the constraint function and resolves the constraints by, essentially, expanding the grammar so that the terms of the constraint function are embedded in the stochastic productions of the grammar. The result is a much larger SCFG grammar, but one that allows us to easily generate, parse, or perform optimal prediction on the artificial language. SLG is described in more depth in Appendix B.

The following is a very much simplified version of the Penglish grammar. The probabilities and constraints have been removed, as have most of the nouns and verbs, depth constraints, and other implementational details. The purpose of providing this simplified grammar is to give the reader a useful frame of reference for the following discussion of the details of the language. A more complete and accurate version of the grammar is shown in Appendix D, along with the lexicon and associated probabilities.

5.2.1 The basic context-free grammar

```
S:       CLS_M "." |
         CLS_M CC CLS_M "."  |
         CLS_S CLS_M "." |
         CLS_M CLS_S ".";

CLS_M:   NP VP;
CLS_S:   SUB_C CLS_M;
CC:      and | but;
SUB_C:   if | because | while | although;

SC:      THAT CLS_M;

RC:      RP IVP | IVP_R |
         RP TVP | TVP_R |
         RP SVP | SVP_R |
         RP PVP | PVP_R |
         RP OBR | OBR;

NP:      SNP | SNP NPP | SNP RC;
SNP:     ART ADJ NN;

VP:      IVP | TVP | SVP | PVP;

IVP:     VERB_I         VPP ADVB;
IVP_R:   VERB_IR        VPP ADVB;
TVP:     VERB_T   NP    VPP ADVB |
         VERB_G   NP NP     ADVB;
TVP_R:   VERB_TR  NP    VPP ADVB |
         VERB_GR  NP NP     ADVB;
SVP:     VERB_S      SC     ADVB |
         VERB_ST  NP SC     ADVB;
```

```
SVP_R:   VERB_SR     SC     ADVB |
         VERB_STR NP SC     ADVB;
PVP:     VERB_P         VPP ADVB |
         VERB_P   by NP VPP ADVB;
PVP_R:   VERB_PR        VPP ADVB |
         VERB_PR  by NP VPP ADVB;
OBR:     NP VERB_T      VPP ADVB |
         NP VERB_G   NP     ADVB;

ADVB:    "" | yesterday | tomorrow | quickly | fiercely | loudly;
VPP:     "" | of NP | on NP | to NP | for NP | with NP | in NP;
NPP:     of NP | for NP | by NP | with NP | on NP;

THAT:    "" | that | if;
RP:      that | who | which;
ART:     "" | a | the | this | these | that | those | some;
ADJ:     nice | mean | fierce | small | big | loud | hard | new | old | young;

NN:      I | we | boy | boys | dog | dogs;

VERB_I:  TELL_AS | TELL_AP | TELL_AI |
         GIVE_AS | GIVE_AP | GIVE_AI;
VERB_IR: TELL_AR | GIVE_AR;

VERB_T:  TELL_AS | TELL_AP | TELL_AI |
         GIVE_AS | GIVE_AP | GIVE_AI;
VERB_TR: TELL_AR | GIVE_AR;

VERB_G:  GIVE_AS | GIVE_AP | GIVE_AI;
VERB_GR: GIVE_AR;

VERB_S:  TELL_AS | TELL_AP | TELL_AI;
VERB_SR: TELL_AR;
VERB_ST: TELL_AS | TELL_AP | TELL_AI;
VERB_STR: TELL_AR;

VERB_P:  TELL_PS | TELL_PP | TELL_PI |
         GIVE_PS | GIVE_PP | GIVE_PI;
VERB_PR: TELL_PR | GIVE_PR;

TELL_AS: told | was telling | had told | tells | is telling | has told |
         will tell;
TELL_AP: told | were telling | had told | tell | are telling | have told |
         will tell;
TELL_AI: told | was telling | had told | tell | am telling | have told |
         will tell;
TELL_AR: telling;

TELL_PS: was told | had been told | is told | has been told |
         will be told;
TELL_PP: were told | had been told | are told | have been told |
         will be told;
TELL_PI: was told | had been told | am told | have been told |
         will be told;
TELL_PR: told;

GIVE_AS: gave | was giving | had given | gives | is giving | has given |
         will give;
GIVE_AP: gave | were giving | had given | give | are giving | have given |
```

```
        will give;
GIVE_AI: gave | was giving | had given | give | am giving | have given |
        will give;
GIVE_AR: giving;

GIVE_PS: was given | had been given | is given | has been given |
        will be given;
GIVE_PP: were given | had been given | are given | have been given |
        will be given;
GIVE_PI: was given | had been given | am given | have been given |
        will be given;
GIVE_PR: given;
```

5.2.2 Main and subordinate clauses

Penglish sentences can consist of a single matrix clause, two coordinate clauses joined by *and* or *but*, or a main clause preceded or followed by a subordinate clause. Four subordinating conjunctions are used, including *if*, *because*, *while*, and *although*. No pauses or punctuation are contained in the sentences to mark clause boundaries. The frequency of occurrence of the various structures is roughly based on the Penn Treebank. Single matrix clauses occur 85.0% of the time and coordinate clauses 9.0%. More than two main clauses are not allowed. Among coordinating conjunctions, *and* and *but* occur 65% and 35% of the time, respectively.

The subordinating conjunctions *if* (2.3%) and *because* (2.2%) are more frequent than *while* (0.7%) and *although* (0.8%). *While* and *although* are slightly biased and *if* is strongly biased toward subordinate-first ordering. *Because* is slightly biased toward subordinate-last ordering. These numbers differ somewhat from those presented in Section 4.7 because they only consider top-level clauses and are taken from just the Brown corpus because of tagging ambiguities in the WSJ.

In order to better model English, tense constraints are placed on a main clause by its subordinate clause for some conjunctions. *While* requires that both the subordinate clause and the matrix clause share the same tense (past, present, or future), with the exception that a future main clause is allowed with a present subordinate clause, as in, "*While the dog sleeps the cat will play*." *If* requires that the main clause be in future tense, with the exception that a present tense main clause is allowed with a present tense subordinate clause, as in, "*If the mother walks away, the baby cries*."

5.2.3 Noun phrases

In addition to the head nouns, NPs in Penglish can contain articles, adjectives, prepositional phrases (PPs) and relative clauses (RCs). PPs and RCs contain other NPs. In order to limit the complexity of Penglish, a maximum depth was placed on the self-embedding of noun and verb phrases. An `NP1` is the highest level NP, used for subjects of sentences with a single main clause. An `NP2` is used as the subject of the main or subordinate clauses in complex sentences or as the subject in an object-extracted RC modifying an `NP1`. An equivalent `OP2` symbol is used for objects of matrix verbs. These level-2 NPs cannot have embedded within them an `NP1`, `NP2`, or `OP2`.

The `NP3`, `OP3`, and `IO3` symbols produce noun phrases that act as subjects, objects, or indirect objects embedded within level-2 NPs. Level-3 NPs are limited to simple noun phrases. They may contain adjectives, but they cannot contain PP or RC post-modifiers. The use of distinct levels of NP prevents infinite recursion but allows a reasonable degree of complexity. Modeling self-embedding with a simple context-free stochastic rule that permits an additional level of embedding modifying any noun with a fixed probability is not an adequate model of English. A single embedding is very common, while a triple embedding is disproportionately rare. Therefore, it was eliminated altogether in Penglish.

Determiners

Most noun phrases begin with a determiner, which includes both articles and quantifiers, of which Penglish uses a limited set. Singular nouns in Penglish can have the articles *a*, *the*, *this*, or *that*. Plural nouns can use *the*, *these*, *those*,

	Singular				Plural				Mass			
Determiner	**S.U.**	**S.M.**	**O.U.**	**O.M.**	**S.U.**	**S.M.**	**O.U.**	**O.M.**	**S.U.**	**S.M.**	**O.U.**	**O.M.**
-none-	—	—	—	—	72.0%	69.1%	82.2%	77.7%	72.0%	70.0%	82.0%	78.0%
the	79.4%	70.9%	42.3%	39.7%	21.0%	27.7%	14.2%	19.9%	21.5%	25.0%	14.4%	20.4%
a/some	14.3%	26.1%	51.8%	59.1%	3.3%	1.7%	1.6%	1.7%	—	—	—	—
this/these	4.5%	2.4%	4.5%	0.9%	2.9%	0.9%	1.4%	0.3%	4.0%	3.0%	2.4%	0.7%
that/those	1.8%	0.6%	1.4%	0.3%	0.8%	0.6%	0.6%	0.4%	2.5%	2.0%	1.2%	0.9%

Table 5.1: Frequency of determiner use in Penglish as a function of number (singular, plural, or mass), role (subject or object), and post-modification (unmodified or modified).

some, or no determiner. Mass nouns can use *the*, *this*, *that*, or no determiner. The pronouns *I*, *me*, *we*, *us*, and *you* don't use determiners. *This* and *that* were included for some variety and to increase the number of roles played by the word *that* to add lexical ambiguity. A larger set of determiners was not included because they were not necessary for any planned experiments.

The probability of nouns being preceded by various articles depends on the role of the noun, whether it is a subject or object, and on whether the noun is post-modified by a PP or RC. Table 5.1 shows the actual frequencies used in Penglish. These are based on the counts that were discussed in Section 4.5. The frequencies for singulars and plurals were drawn from the Treebank corpus, while those for mass nouns were estimated, mostly based on plural usage.

Adjectives

Penglish uses a limited set of adjectives, all of which are descriptive. These include *nice*, *mean*, *fierce*, *small*, *big*, *loud*, *hard*, *new*, *old*, and *young*. The adjectives *nice* and *hard* each have two different meanings, which depend on the modified noun. This set of fairly common, simple adjectives was chosen because they can each be used to modify a number of different nouns in the lexicon. An adjective like *steep*, on the other hand, would be less productive.

In Penglish, one adjective at most is allowed to modify a noun. The frequency of modification is dependent on whether the NP is a subject or object and on whether the NP is post-modified by a PP or RC. These frequencies are based on the analysis reported in Section 4.5.2. The frequency of adjective modification is as follows: modified objects, 32.5%; unmodified objects, 23.3%; modified subjects, 23.5%. The probability of adjective modification of a non-post-modified subject was supposed to be 10.0%, but due to a typographical error it was set to 1.0%. Therefore, experimental results dependent on subject-modifying adjectives in Penglish are questionable.

Post-modification

Nouns in Penglish can be post-modified by either a PP or an RC, but not by more than one post-modifier. The frequency of post-modification is dependent on whether the NP is a matrix subject, a matrix object, or embedded within a PP, RC, or sentential complement (SC). The frequency of post-modification was taken directly from the results reported in Table 4.16, with the possibility of modification by both a PP and RC removed and the probability of the remaining options renormalized. The contents of the RCs and PPs are discussed in Sections 5.2.5 and 5.2.6.

5.2.4 Verb phrases

Verb phrases are the most complex aspect of the Penglish grammar. In generating verb phrases, one must consider a number of properties, including the root verb itself, its argument structure, tense, aspect, and voice, and number and semantic agreement with the subject and objects. When generating a verb phrase, each of these properties represents either a decision that must be made or a constraint, based on earlier decisions, that must be obeyed. What will the verb be? Which argument structure will it use? How can we make sure that the verb and subject agree in number and that the tense of the verb agrees with the tense of another verb in a subordinate clause?

In designing a context-free grammar, it is not practical to make all of these decisions at once. The designer must choose how to structure the grammar so that the decisions are made in a practical order that enables the easy application of the necessary constraints. The method used in the Penglish grammar is as follows. First the major argument structure and voice are chosen, the options being intransitive, transitive, transitive with a sentential complement, and passive. Then the sub-argument structure is chosen. Penglish allows single-object transitives, but not ditransitives, to have an optional PP. Verbs with sentential complements can have either no other objects or an indirect object, as in, "*I told Duane he was wrong.*" All passives can include an optional "by-phrase" expressing the agent.

Each of these six argument structures (intransitive, transitive, ditransitive, SC, SC with an object, and passive) generates a subset of symbols that only produce verb forms capable of using that argument structure. Each of these verb forms generates phrases using a single verb with a single form of agreement. For example, in the simplified Penglish grammar shown earlier, `TELL_AS` produces active, singular forms of the verb *tell* (*is telling*). `TELL_AP` produces active, plural forms (*are telling*) and `TELL_AI` produces active, first-person, singular forms (*am telling*). `TELL_PS`, `TELL_PP`, and `TELL_PI` produce passive forms. `TELL_AR` and `TELL_PR` produce active and passive forms used in reduced relative clauses, to be discussed in Section 5.2.5.

Each of these verb form sub-symbols produces a number of phrases that differ in tense and aspect. For example, `TELL_AS` generates the phrases *told*, *was telling*, *had told*, *tells*, *is telling*, *has told*, and *will tell*. In summary, the order in which properties of verb phrases are chosen in this grammar is basic argument structure and voice, specific argument structure, verb base form and number agreement, and then tense and aspect.

At each stage in this process, the probabilities of making every possible decision must be specified in the grammar. These probabilities were determined using a lexically-driven approach. A number of statistics were gathered for each verb in the lexicon. These values are described in more detail in Section 5.3, but they include such measures as the frequency with which each verb occurs in each argument structure and tense. The probability of producing an intransitive argument structure was determined using the aggregate frequency with which all verbs occur in the intransitive. The probability of producing a particular verb given an intransitive argument structure is determined using the ratio of the frequency of that verb in the intransitive to the frequency of all verbs in the intransitive. Needless to say, these probabilities were not computed by hand. A program was written to take the lexicon with associated frequencies, determine the corresponding structural probabilities, and then insert those probabilities into the context-free grammar.

One additional complexity in generating verb phrases which is not included in the simplified grammar shown in Section 5.2.1 is the need to track the depth of embedding to prevent infinite recursion. As with the NPs, there is a different version of each VP for every depth of embedding. A `TVP1` is a transitive VP used in single matrix clauses. Its direct object is an `OP1`, since it is not embedded. In the case of a ditransitive, the indirect object is an `OP2`, which limits its complexity. A `TVP2` is used inside subordinate or coordinate clauses and top-level RCs and SCs, which are directly contained within the matrix clause. Its objects are `OP2`s, which can contain just one more level of embedding, and its indirect objects are `IO3`s (equivalent to `OP3`s) which cannot be post-modified. Finally, a `TVP3` occurs within a nested RC or SC. It only produces `OP3`s and `IO3`s, which permit no further post-modification.

Sentential complements

The implementation of sentential complements is relatively simple in Penglish. Only verb-modifying SCs are allowed. An SC consists of an optional complementizer, which could be either *that* or *if*, followed by a main clause having a subject and predicate. The frequency with which *that*, *if*, or no complementizer is used depends on the verb. For example, the verb *ask* requires an *if*. The verb *know* uses *if* 26.7% of the time and *that* 53.5% of the time. Most other verbs cannot use the complementizer *if*. The frequency of an SC being reduced ranges from about 12% for *see* to 91% for *wish*.

Post-modification

Verbs in Penglish can receive post-modification by adverbs, prepositional phrases, or both. All verbs can be modified by an adverb, but the adverb always follows the verb and all other objects and modifiers. Prepositional phrases are also optional, but will immediately precede any adverb. The same word cannot be modified by more than one PP. Additionally, prepositional phrases are not allowed with some complex argument structures, including verbs with ditransitives and sentential complements. The reason for this was just to limit the complexity of the language, but this seems to have been a questionable decision.

Penglish uses just five adverbs, two of them temporal and three descriptive: *yesterday*, *tomorrow*, *quickly*, *fiercely*, and *loudly*. The adverbs that are able to modify each verb and the overall probability of adverb modification was set on a verb-by-verb basis based on corpus counts. The probability that the verb is modified by an adverb is simply distributed evenly among the available adverbs. The use of the temporal adverbs is further constrained by the tense of the verb. *Tomorrow* is only allowed with future verbs while *yesterday* is only allowed with past-tense verbs.

5.2.5 Relative clauses

Because of the interest in relative clauses in the empirical literature and the range of factors believed to be relevant to their difficulty, some care was taken in the generation of relative clauses. Section 5.2.3 discussed the probability that a particular noun is modified by an RC. In this section we consider the contents of those RCs.

Five main types of RCs are used in Penglish, including subject-extracted intransitives, transitives, and sentential complements, object-extracted transitives (object-relatives), and passives. Each of these RC types can also be reduced or marked with a relative pronoun (*that*, *who*, or *which*). Reduced object-relatives are identical to marked ones aside from the missing pronoun. Reduced passives are also missing the auxiliary verb. Reduced subject-relatives are limited to the present participle form, as in, "*The gnat biting my arm has but a moment to live*." Therefore, for all of the reduced RCs other than object-relatives, a special form of the verb was required.

The frequency with which the five types of RC occur in their reduced and unreduced forms in Penglish is dependent on the type of noun under modification. Separate probabilities are used for matrix subjects, matrix objects, and all other nouns. These probabilities are exactly those derived from the Treebank and shown in Tables 4.18, 4.19, and 4.20.

5.2.6 Prepositional phrases

Both nouns and verbs can be modified by prepositional phrases in Penglish. The frequency of noun modification is dependent on whether the noun is a matrix subject, matrix object, or plays some other role. These frequencies were given in Table 4.16. The frequency of verb modification is verb-dependent and is based on lexical frequencies. Each noun and verb constrains the types and frequencies of the PPs that can be used to modify it, as well as the nouns used in the object of the preposition.

A number of different verb-modifying prepositional phrases (VPPs) are used in Penglish. These are distinguished by their semantic role, and not necessarily by their prepositions. VPPs can express theme (*heard of the boy*), recipient (*gave the ball to the boy*), beneficiary (*read the book for the boy*), instrument (*ate fish with a knife*), destination (*leave for the park*; *drove to the zoo*), enclosing location (*left the book in the park*), and surface location (*left the book on the table*).

Noun-modifying prepositional phrases (NPPs) have a different set of roles but share some of the same prepositions. They are used to express subtype (*the manager of the school*; *the school for girls*), beneficiary (*the apple for the teacher*), author (*the book by the player*), possession (*the girl with a ball*), accompaniment (*the girl with the cop*), and location (*the bird in the park*). Thus, most, but not all, prepositions modify both nouns and verbs and some, such as *with* or *for*, can play multiple roles that are dependent on the modified word.

5.3 The lexicon

Previous artificial training languages used in connectionist models have had very limited vocabularies. In order to perform the intended experiments in this study, a significantly larger set of nouns and verbs is necessary. This section discusses some of the properties of the nouns and verbs in Penglish. Further details of the lexicon are given in Appendix D.

5.3.1 Verbs

A variety of considerations went into the choice of lexical items, particularly for verbs. To begin with, there must be enough verbs with each type of argument structure. This includes obligatorily transitive, obligatorily intransitive, and mixed transitivity verbs, as well as ditransitives. There must also be verbs that can take sentential complements, with and without an additional object. In order to model the results on sentential complements discussed in Section 3.4, we need a set of verbs capable of using an SC that have a high probability of SC reduction and another set with a low probability of reduction. In order to model the results on the main verb/reduced relative ambiguity discussed in Section 3.3, we will need verbs that vary in their frequency of occurrence in passive constructions, in passive relative clauses, and in reduced relatives. Furthermore, some of the verbs must have distinct past tense and past participle forms and others must have identical forms.

Countering this need for a diverse set of verbs is the consideration that the larger the lexicon, the harder the language will be for the network to learn, possibly resulting in either poor performance or overly long training, given computational limitations. Ultimately, 56 verb roots were selected. These include 44 different verbs, as well as 11 verbs with multiple senses (three senses of *know* were used, none of them biblical). Subtle distinctions between verb senses were not made in the language, but broader distinctions were drawn, often reflecting senses that differed in argument structures. These distinctions are drawn sometimes in syntax and sometimes only in semantics.

For example, there is a difference between a person *driving* a car and a car *driving*. Similar distinctions are drawn for the verbs *fly*, *hit*, and *play*. There is also a difference between *forgetting* an object (leaving it behind) and *forgetting* a fact, and between *leaving* a place and *leaving* an object behind. The communication verbs *write*, *tell*, and *ask* have different forms for *writing* (to) a person and *writing* a book. The verb *know* has three forms, corresponding to *knowing* a fact, *knowing* a person, animal, or object, and *knowing* of or about something.

Each verb or verb sense has its own entry in the lexical database, more details of which are shown in Appendix D.2 and a summary of which is shown in Table 5.2. The entry begins with the five forms of the verb: its past tense, singular and plural present tense forms, and present and past participles. The remainder of the entry is composed mainly of occurrence counts derived from the analysis of verb phrases in the Penn Treebank that was discussed in Chapter 4.

The first set of counts represent the frequency of verb use in a variety of argument structures and related contexts. These include the number of occurrences with no object, a single object, a double object dative, a prepositional dative, an unreduced SC, a reduced SC, an unreduced SC with an object, a reduced SC with an object, an unreduced passive RC with and without a by-phrase, and a reduced passive RC with and without a by-phrase.

The next set of values reflects occurrences in various tenses and voices, including simple active past tense, past progressive, past perfect, simple future, and simple, progressive, and perfect present tense. Also included are counts for the passive tenses used in Penglish: past, past perfect, present, present perfect, and future. The third set of values indicate the frequency with which the verb is modified by each of the prepositions, *of*, *for*, *with*, *to*, and *on*.

The remaining parts of a verb's lexical entry are specified by hand, rather than through corpus analysis. First is whether the verb can use *that*, *if*, or both as a complementizer, assuming it takes an SC. The next set of fields define the verb's thematic roles and the nouns that can fill those roles. For example, the subject of *take* must be either a human or an animal and it serves as the agent. The verb *read*, on the other hand, must have a human as its subject and it serves as the experiencer.

Besides the subject, the other modifiers whose roles and possible nouns are defined in the lexical entry are the direct object, indirect object, and the objects of the seven verb-modifying PP types. If the verb cannot have a modifier of a certain type, the entry is left blank. Note that, like the subject, PPs headed by a certain preposition can play different thematic roles for different verbs. The final part of the entry specifies the adverbs that can modify the verb.

5.3.2 Nouns

Penglish uses 45 noun roots, three of which are secondary senses of the same surface form. Some properties of these nouns are shown in Table 5.3. There is a flying *bat* and a wooden *bat*. The word *baseball* can refer to the ball itself or to the game in general and the word *school* can refer to a specific school (*The school burned down.*) or to the institution in general (*Graduate school is fun—for the first nine years.*).

Verb	*Intransitive*	*Transitive*	*Ditransitive*	*SC*	*SC w/ object*	*Agent*	*Experiencer*	*Patient*	*Theme*	*Source*	*Instrument*	*Beneficiary*	*Location*	*Goal*	*Companion*
ASK_P	○	○				●				○		○	○		
ASK_Q		○		○	○	●		○		○			○		
BELIEVE	○	○		○	○	●				○					
BARK	●					●									
BITE	○	○				●		○							
BUY	○	○	○			●			○			○			
CONSIST	●					●			●						
DRIVE_H	○	○				●		○				○		○	
DRIVE_C	○	○				●		○				○		○	
EAT	○	○				●		○			○		○		
EXAMINE		●					●		○		○		○		
FEEL		○		○			●		○						
FIND		○		○	○	●			○		○	○	○		
FISH	●					●			○				○		
FOLLOW	○	○				●			○				○	○	
FORGET_H	○	○		○	○		●		○						
FORGET_L		○		○	○	●			○				○		
FLY_H	○	○				●		○						○	
FLY_P	●					●							○	○	
GET		○	○			●			○			○	○		
GIVE	○	○	○			●			○			○	○	○	
GO	●					●							○	○	○
GUESS	○	○		○		●		○							
HAVE		●				●			○			○	○		
HEAR	○	○		○			●		○				○		
HIT_H	○	○				●		○			○	○	○	○	
HIT_O	○	○				●		○					○		
HOPE	○			○			●		○						
INVOLVE		●				●			●						
KILL_H	○	○				●		○			○		○		
KILL_A	○	○				●		○					○		
KNOW_P		●					●		●						
KNOW_O	●						●		●						
KNOW_T	○	○		○			●		○						
LEAVE_I	○	○				●				○				○	
LEAVE_T		●				●			○				○		
PARK	○	○				●		○					○		
PLAY_T		○				●			○				○		
PLAY_I	●					●							○		○
PUT		●				●			●				●		
QUESTION		○		○		●		○			○	○	○		
READ	○	○	○	○	○		●		○		○	○	○	○	
REALIZE				○			●								
SAY		○		○		●			○			○	○	○	
SEE	○	○		○			●		○		○		○		
SHOW		○	○	○	○	●	○		○			○	○		
TAKE		○	○			●			○			○	○	○	
TELL_S	○	○	○	○	○	●			○				○	○	
TELL_P		●				●			○				○	●	
THINK	○	○		○			●	○	○				○		
THROW	○	○	○			●		○				○	○	○	
USE		●				●		○					○		
WANT		●				●			●			●	●		
WISH				●		●									
WRITE_S		○	○	○	○	●		○			○	○	○	○	
WRITE_P		●				●					○	○	○	○	

Table 5.2: Summary of the argument structures and roles used by the Penglish verbs. Filled circles are required, empty ones are optional.

Noun	*Singular*	*Plural*	*Human*	*Animal*	*Object*	*Place*	*Subtype*	*Beneficiary*	*Possession*	*Location*	*Companion*	*Author*
I	∘	∘	∘									
YOU		∘	∘									
BOY	∘	∘	∘				∘		∘	∘	∘	
GIRL	∘	∘	∘				∘		∘	∘	∘	
LAWYER	∘	∘	∘				∘		∘	∘	∘	
TEACHER	∘	∘	∘				∘		∘		∘	
REPORTER	∘	∘	∘				∘		∘	∘	∘	
PLAYER	∘	∘	∘				∘		∘		∘	
COP	∘	∘	∘						∘		∘	
FATHER	∘	∘	∘				∘		∘	∘	∘	
MOTHER	∘	∘	∘				∘		∘	∘	∘	
MANAGER	∘	∘	∘				∘		∘		∘	
OWNER	∘	∘	∘				∘		∘		∘	
CAT	∘	∘		∘					∘	∘	∘	
DOG	∘	∘		∘					∘	∘	∘	
BIRD	∘	∘		∘					∘	∘	∘	
FISH	∘	∘		∘					∘	∘		
BAT_A	∘	∘		∘					∘	∘		
QUESTION	∘	∘					∘	∘		∘		∘
ANSWER	∘	∘										∘
TEST	∘	∘					∘	∘				∘
STORY	∘	∘					∘	∘	∘	∘		∘
BOOK	∘	∘			∘		∘	∘	∘	∘		∘
APPLE	∘	∘			∘			∘		∘		
FOOD	∘				∘			∘	∘	∘		
CAR	∘	∘			∘			∘	∘			
PLANE	∘	∘			∘			∘	∘			
BALL	∘	∘			∘			∘		∘	∘	
BAT_O	∘	∘			∘			∘		∘	∘	
BASEBALL_B	∘	∘			∘			∘		∘	∘	
BASEBALL_G	∘											
TRUMPET	∘	∘			∘			∘		∘		
EVIDENCE	∘				∘		∘	∘		∘		
SOMETHING	∘			∘	∘			∘		∘		∘
BINOCULARS		∘			∘			∘		∘	∘	
KNIFE	∘	∘			∘			∘		∘		
PEN	∘	∘			∘			∘		∘	∘	
PICTURE	∘	∘			∘		∘	∘	∘	∘		∘
TABLE	∘	∘			∘		∘		∘	∘		
FLOOR	∘											
HOUSE	∘	∘				∘	∘		∘			
PARK	∘	∘				∘	∘		∘			
ZOO	∘	∘				∘	∘		∘			
SCHOOL_I	∘	∘				∘	∘		∘			
SCHOOL_G	∘					∘						

Table 5.3: Summary of the noun types and possible roles played by prepositional phrases modifying the noun. Note that these are not the semantic encodings of the nouns.

The lexical entry for a noun is simpler than that for a verb. It begins with the singular and plural forms of the noun in the nominative and accusative, although only the pronoun *I* has different accusative forms. Next is the frequency of the noun in the singular and plural. Mass nouns, like *food*, only have singular forms and other words, like *binoculars*, only have plural forms. The noun *fish* is special in that its singular and plural are identical. The pronoun *you* in English, other than Southern or Pittsburghian English, is the same in both the singular and plural forms. However, *you* always uses plural verb forms. In Penglish, *you* is always treated as a plural to avoid global ambiguities in meaning.

The next set of values in the lexical entry are the frequencies with which the noun is modified by the prepositions *of*, *for*, *by*, *with*, and *on* in the Treebank. Next are the adjectives that can modify the noun. The final set of fields defines the nouns that can serve as the objects of each type of noun-modifying PP and the role played by that PP. For noun modifiers, these roles include subtype, beneficiary, possession, location, companion and author.

5.4 Phonology

Earlier versions of this model used localist encodings of words at the comprehension and production interfaces. Each word was simply represented by the activation of a single unit. This type of encoding has certain advantages if one is interested in purely syntactic processing, since it denies the network any phonological information that might be helpful in interpreting word class. When using a localist input representation, the weight vector projecting out of that word's input unit, following training, can be taken as the network's learned representation of the word's meaning (Elman, 1991). A localist representation is also particularly useful in prediction or production because the network can simply produce a distribution of activation over the output units representing the probability that each word will occur next.

However, there are many disadvantages in using localist word representations. The sound and spelling patterns of words in natural languages do correlate with semantic and syntactic information. Morphological variants of a noun or verb root generally have very similar sound patterns, and are usually identical up to the ending. The inflectional morphemes at the ends of words are strong indicators of their syntactic and semantic roles. Words ending in *-t* or *-ed* sounds are most likely past tense verbs and those ending in *-ing* are most likely present participles. The sound pattern of the words in a language is a strong, but not perfectly reliable, indicator of semantic and syntactic regularities.

A competent language processor should be able to extract useful information from phonology to aid learning and to promote generalization. Therefore, rather than using localist word representations, Penglish uses distributed, featural word representations. Actually, there are two different representations of words, one for comprehension and one for production. Both representations encode an entire word in a single vector, rather than arranging phonemes or syllables temporally.

The comprehension, or input, word representation uses a distributed pattern of activation over phoneme units grouped into up to three syllables, which we will call A, B, and C. A word with one syllable just uses part A of the representation. A word with three syllables uses parts A, B, and C. However, a word with only two syllables uses parts A and C, rather than A and B. This was done so that the endings of words will be aligned, whether the words have two or three syllables. The small number of words with more than three syllables were shortened in such a way that they retained their morphological inflections. For example, the word *binoculars* became *bi-noc-lars* and *realizing* became *re-lize-ing*.

Each of the three syllables is composed of an onset, a vowel, and a coda, which use distinct sets of phoneme units. The one-syllable word *schools* consists of the onset sk, the vowel U, and the coda lz. The order of phonemes within the onset or coda is not distinguished. In the case of *schools* the onset is simply represented by activating both the s and k units. However, it should be noted that the unit for s in an onset and s in a coda are distinct and that different syllables use different sets of units. The comprehension representation uses 169 bits, 93 for the syllable A, 23 for the syllable B, and 53 for syllable C.

The representation used in prediction and production is not the same as that used in comprehension. During production, it must be possible to decide which word the network would like to produce at a given point. If production were to use a fully distributed representation, the choice would be very difficult unless the desired word were always much stronger than its nearest competitor. On the other hand, using a completely localist output representation inhibits the network from taking advantage of morphological regularities.

As a compromise, Penglish uses a two-part localist representation for production. The main part of the representation is a localist encoding of the word stem. Two words are given the same stem only if they are clearly composed of a similar sounding stem with morphological variation. Verbs with irregular past tenses involving vowel change have different stems in the present and past tense. For example, *bite* and *bit* have different stems. On the other hand, *fish* the noun and *fish* the verb share the same stem, as do *park* the verb and *park* the noun.

The second part of the output representation is a localist encoding of only the most common inflectional morphemes. These include -d, -ing, -ly, -n, -o, -z, and no morpheme. For example, the words *bark*, *barked*, *barking*, and *barks* are encoded bark, bark -d, bark -ing, and bark -z. Although some regular past tense verbs end in the t sound and others in the d sound, this distinction is predictable from context and is relatively transparent. Therefore, no distinction is made between the two in this representation and -d is used for both.

5.5 Semantics

As important as the syntax of Penglish is the way in which its semantics are represented. The semantic representation is meant to encode the meaning of the sentence, which is referred to here as its *message*. Ideally, messages should be language independent, meaning that multiple languages with different syntax should share the same semantic representations. While this ideal may not be perfectly realizable in natural language, it should be possible for multiple languages to share substantially similar messages.

For a skilled comprehender, the meaning of a sentence may actually have multiple levels of inferential abstraction. At the lowest level, only the relationships between the literal components of the sentence are encoded. At a somewhat higher level, implied constituents might be added. To borrow an example from St. John and McClelland (1988), if the sentence states that someone ate soup, it may be reasonable to infer that a spoon was used, even if it is not literally stated. At an even higher level are more complex inferences. If "John lost control and hit a tree on the way home," reasonable assumptions are that he was probably driving a car and that he, the car, and the tree may be in bad shape. Even deeper levels of inference enable the use of innuendo and sarcasm. Nevertheless, comprehension in the CSCP model is limited to deriving the lowest, literal, representation of sentence meaning.

Most earlier comprehension models were trained only on simple sentences. This enabled the use of semantic representation with a fixed number of slots to store the verb and the constituents filling its thematic roles. A static, fixed-length representation of sentence meaning such as this is helpful for two reasons. In production, we need the message to serve as an input to the production system. In comprehension, it is more efficient to learn with a static message representation serving as a target than it is to learn with something dynamic.

Unfortunately, with the complex sentences possible in Penglish, it would not be practical to use a slot-filler representation to handle the meaning of all possible sentence structures. A noun serving a particular thematic role must be tied to a particular verb, and relationships must be formed between multiple clauses. To do this in a hand-designed static representation would require a tremendous number of slots. Therefore, a more sophisticated message representation is necessary.

One reasonable approach to the semantic encoding of complex sentences is to decompose the meaning of the sentence into a number of propositions. Each proposition can express the relationship between a head and a modifier, or between two heads. These propositions might be treated independently or organized into some sort of hierarchical structure. But in order to achieve the goal of having a static representation of sentence meaning for comprehension and production, there must be a way to compress the propositional structure into a fixed-length vector. The approach taken in the current model will be to use a separate *encoder* neural network, which learns to encode the propositional meaning structure into static messages and decode messages into their propositional structure.

5.5.1 Semantic trees

The initial idea in developing the representation of sentence meaning used in the current model was that semantic relationships could quite naturally be encoded as a tree structure. Different from a syntactic tree, the *semantic tree* developed for an earlier version of the model was a ternary tree in which one child of a non-terminal expresses the relationship that exists between the other two children. In the original design of the tree, the head of the left branch

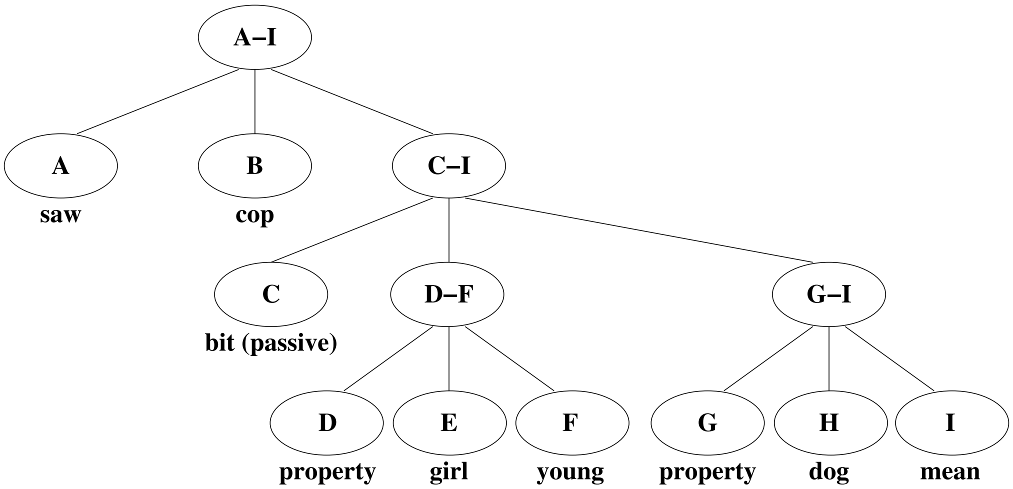


Figure 5.1: A semantic tree for the sentence, "*A cop saw the young girl that was bitten by that mean dog*."

of a non-terminal represented the relationship between the head of the middle branch and the head of the right-most branch. The head of a subtree was the terminal reached by following its middle branches. Thus, heads were in the middle and relationships were on the left. A verb served as the relationship between its subject and object, making its subject noun the head of a sentence.

Figure 5.1 shows how the sentence, "*A cop saw the young girl that was bitten by that mean dog*," might be represented in this sort of tree. The subject, *cop* is the head of the main tree. *girl* is the head of its right child, C-I, because *girl* is its center-most descendant. Thus, node A-I expresses the fact that the cop saw the girl. The relationship expressed by node C-I is that the girl was bitten by the dog. Finally, the relationships expressed by nodes D-F and G-I are that the girl was young and the dog was mean.

Aside from its apparent elegance, one reason for using a tree with a fixed branching factor for representing sentence meaning is that a special form of an auto-association network, known as a recursive auto-associative memory, or RAAM, can be used to compress trees of this sort into fixed-length, static representations (Pollack, 1988, 1990). Essentially, the RAAM starts with a non-terminal at the bottom of the tree, whose children are all terminals. The RAAM compresses the vectors representing those three terminals into a single vector of the same length as the original three. It then replaces the subtree with the new vector, decreasing the size of the tree. Eventually, the whole tree has been replaced by a single, static vector, which can be treated as the message of the sentence for use in either comprehension or production.

Although the RAAM method is reasonably effective at encoding and decoding trees of this sort, it does not provide a very good foundation for comprehension and production. The RAAM has the luxury of using many recursive processing steps to compress the tree or to expand the compressed tree into its original form. As a result, the compressed representations that it forms tend to be extremely complex. Information about the structure of the original tree or properties of the terminals in the tree are buried deep in this convoluted representation. This creates a problem because the comprehension and production network must try to encode or decode that message vector in just a few steps as each word comes in or goes out. As a result, RAAM-encoded messages do not make good targets for comprehension or good sources for production.

Aside from the issue of how they are compressed, tree-based representations of semantics have shortcomings of their own. One problem with using a tree structure like the one in Figure 5.1 is that there is only a minimal difference between the encoding of a passive, *the girl was bitten by the dog*, and the active form with a very different meaning, *the girl bit the dog*. The only difference between these two is in the encoding of the verb.

A more serious problem with this design was revealed when sentential complements were added to the language. Consider the slightly different sentence, "*A cop saw that the young girl was bitten by that mean dog*." One possibility is to represent the sentence with a tree identical in structure to Figure 5.1. In this case, the only way to distinguish the sentence containing a relative clause from the one containing a sentential complement would be to change a few bits

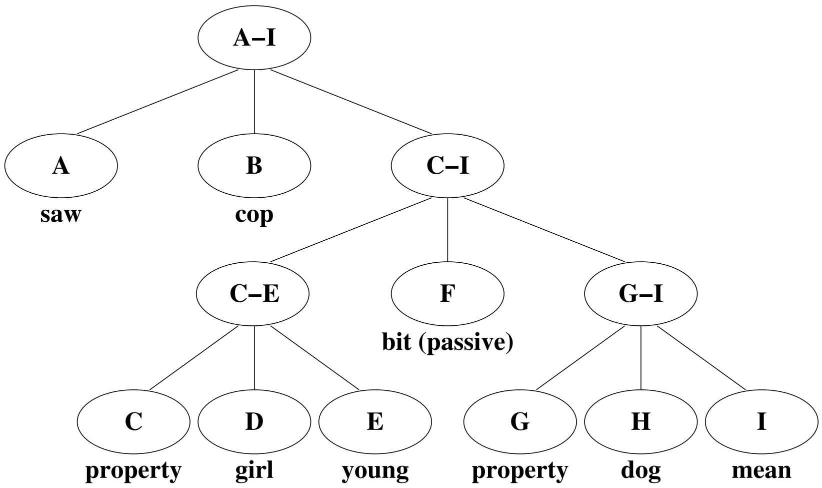


Figure 5.2: A semantic tree for the sentence, "*A cop saw that the young girl was bitten by that mean dog*."

somewhere, perhaps in the representation of *the girl* or *was bitten*. This does not seem quite satisfactory since there really is quite a big difference in meaning between the two sentences. In the sentential complement sentence, the cop is really seeing the whole action of the biting, not just the girl. Therefore, perhaps the head of the subtree ought to be *was bitten*, rather than *the girl*. An alternative semantic tree is shown in Figure 5.2. The problem with this tree is that *was bitten* is now at the center of subtree C-I and is no longer in position to express the relationship between *the girl* and *the dog*.

This is just one example of the problems that can arise in trying to develop a ternary tree structure for representing complex sentences that include actives and passives, relative clauses, sentential complements, subordinate clauses, and multiple thematic roles beyond just agents and patients. Eventually I decided that it is not possible to produce a coherent tree representation of sentence meaning. A more flexible approach was needed.

5.5.2 Proposition sets

A simpler and more flexible design for a semantic representation is simply to use a set, or collection, of propositions. In the earlier language, the verb, or *action*, served as the relationship between its subject and object. However, with the addition of prepositional phrases, Penglish has a more sophisticated set of thematic roles. Therefore, a new representation is now used, in which the different thematic roles serve as relationships between the action and just one of its constituents. Thus, subjects and objects are treated just like the other verb modifiers. This also enables subjects and objects to use different thematic roles for different verbs. For example, the clause, "*The cop saw the girl,*" is now encoded with two propositions, one stating that the experiencer of the seeing action is the cop and the other stating that the theme of the seeing action is the girl.

In this proposition-set representation, the meaning of the sentence, "A cop saw that the young girl was bitten by that mean dog," is encoded using the six propositions shown in Table 5.4. The first states that the experiencer of the seeing act is an indefinite cop. The second states that the sentential complement argument of the seeing act is the biting act. The third states that the patient of the biting is the girl, but in this case the EMPHASIS bit is on, which means that the focus in this action is on the girl. The EMPHASIS bit is discussed further in Section 5.5.4. Because the patient has the focus, this translates into a passive construction. The fifth proposition states that the agent of the biting is the dog. The fourth and sixth simply state that the girl is young and the dog is mean.

It is interesting to contrast this with the case of the sentence shown in Figure 5.5, which uses a relative clause rather than a sentential complement. Two of the propositions have changed. The second proposition now states that the theme of the seeing is the girl and the third states that the patient of the biting is the girl, but that this action is expressed as a restrictive relative clause.

SENSE SEE PAST SIMPLE	EXPERIENCER EMPHASIS	LIVING HUMAN BIG OLD MEAN MALE OCCUPATION COP A
SENSE SEE PAST SIMPLE	SC THAT	ACTION AFFECTING BITE PAST SIMPLE
ACTION AFFECTING BITE PAST SIMPLE	PATIENT EMPHASIS	LIVING HUMAN SMALL YOUNG FEMALE THE
LIVING HUMAN SMALL YOUNG FEMALE THE	PROPERTY	PROPERTY GOOD YOUNG
ACTION AFFECTING BITE PAST SIMPLE	AGENT	LIVING ANIMAL PET FIERCE BIG DOG THAT
LIVING ANIMAL PET FIERCE BIG DOG THAT	PROPERTY	PROPERTY BAD MEAN

Table 5.4: Propositional representation of, "*A cop saw that the young girl was bitten by that mean dog.*"

SENSE SEE PAST SIMPLE	EXPERIENCER EMPHASIS	LIVING HUMAN BIG OLD MEAN MALE OCCUPATION COP A
SENSE SEE PAST SIMPLE	THEME	LIVING HUMAN SMALL YOUNG FEMALE THE
ACTION AFFECTING BITE PAST SIMPLE	PATIENT RC THAT	LIVING HUMAN SMALL YOUNG FEMALE THE
LIVING HUMAN SMALL YOUNG FEMALE THE	PROPERTY	PROPERTY GOOD YOUNG
ACTION AFFECTING BITE PAST SIMPLE	AGENT	LIVING ANIMAL PET FIERCE BIG DOG THAT
LIVING ANIMAL PET FIERCE BIG DOG THAT	PROPERTY	PROPERTY BAD MEAN

Table 5.5: Propositional representation of, "*A cop saw the young girl that was bitten by that mean dog.*"

5.5.3 The syntax/semantics distinction

It is important to note that the semantic representations used in the model are intended to encode sentence meanings, divorced, as much as possible, from the syntax of Penglish, or any other language. Therefore, a distinction is drawn between *verbs* and *actions* and between *nouns* and *objects*. *Verb* and *noun* are syntactic designations of words. *Action* and *object* are classifications of semantic elements. It just so happens that verbs tend to be used to denote actions and nouns tend to be used to denote objects.

Some readers may rightfully object to the use of the term *action* to refer to all things denoted by verbs, since many verbs, obviously, are not action verbs. Similarly, not all nouns denote objects. *Action* and *object*, as used here, are just terms of convenience. Among the class of things lumped together as *actions* are true physical actions (*kill*), mental acts (*think*), sensing acts (*hear*), states (*have*), and combinations of these things. For example, *saying* something involves both mental and physical aspects. Although these things are all called *actions*, there is not necessarily any commonality in their semantic encodings. There is no overlap in the basic representation of *killing* and *thinking*. The only overlap that might occur is if they both occurred in the past tense, for example.

Likewise, the word *object* is used to refer to the meanings of true objects, as well as people, animals, and places. Again, these things do not necessarily have any overlap in terms of their semantic representations. Even in terms of their use there is often little overlap. *Dog* and *park* are both termed *objects* here, but a *park* can only be an agent of the verb *have* and can only be a theme of the verbs *find*, *see*, or *hear*. *Park* is nearly always a location or goal. *Dog*, on the other hand, can never be a location, but is the agent and patient of many verbs.

Therefore, the CSCP model is not being told in any clear way, by virtue of the semantic representations available to it, that there is one set of syntactic elements called nouns and another called verbs. The semantic representation contains no *noun* or *verb* bits. Two nouns can refer to “objects” with disjoint features and two verbs can refer to “actions” with disjoint features. If the model is to learn a noun/verb distinction, it must be sensitive not only to their corresponding semantics but also to their commonalities in usage syntactically.

5.5.4 Proposition encoding

This section describes the featural bits used to encode the three elements of a proposition.

Relationship features

Table 5.6 shows the set of bits used to encode the relationship, or middle element, of a proposition. Adjective and adverb modifiers are encoded with the PROPERTY relation. The right element is the property and the left is either the action or object being modified. The next set of bits represent relationships between the actions in two different clauses. The AND bit encodes a conjunction of actions, or coordinate main clauses. There were supposed to be distinct AND and BUT bits, but due to a bug the AND bit was used for both. Therefore, AND and BUT are synonymous in Penglish. Fortunately, no planned experiments are sensitive to this distinction.

The next four action/action relationships are for subordinate clauses. IF, for example, means that the action on the left is conditioned on the action on the right. BECAUSE means that the action on the left is caused by the action on the right. The EMPHASIS bit can be used in conjunction with either of these and its presence or absence indicates whether the subordinate clause should precede or follow the main clause.

The other action/action relation, SC, indicates that the second action is the head of a sentential complement of the first action.

The largest set of relationship bits is used to encode the roles played by noun modifiers. Ten of these encode thematic roles, or relationships between an action and one of its constituents, and six encode relationships between objects. BENEFICIARY, COMPANION, and LOCATION can encode either action/object or object/object relations.

The decision to use discrete, localist units to represent thematic roles such as agent and patient was one of convenience. It does not reflect any opposition, on my part, to theories that call for a more fine-grained featural encoding of thematic role (Dowty, 1991). I expect that, in transforming these propositional representations, the model will work out its own form of distributed encoding such that agents and experiencers will share a high degree of similarity, as

Left	Relationship Bit	Right
act/obj	PROPERTY	property
action	AND/BUT	action
action	IF	action
action	BECAUSE	action
action	WHILE	action
action	ALTHOUGH	action
action	SC	action
action	AGENT	object
action	EXPERIENCER	object
action	GOAL	object
action	INSTRUMENT	object
action	PATIENT	object
action	SOURCE	object
action	THEME	object
act/obj	BENEFICIARY	object
act/obj	COMPANION	object
act/obj	LOCATION	object
object	AUTHOR	object
object	POSSESSION	object
object	SUBTYPE	object
EMPHASIS, RC, THAT, WHOICH, PP		

Table 5.6: The bits used to encode relationships.

will patients and themes.

The final five relationship bits are modifiers that can appear in conjunction with one of the main relations. EMPHASIS is used for two main purposes. The first, as already mentioned, is to indicate whether a subordinate clause precedes or follows a main clause. Its other use is to denote which thematic constituent is the subject of the clause, and is thus emphasized.

The RC bit is turned on for the action/object relationship when the verb denoting the action heads a relative clause modifying the noun denoting the object. When the relative clause is marked with the relative pronoun *that*, indicating a restrictive sense, the THAT bit is added. If the RC uses *who* or *which*, the WHOICH bit is used. Although in retrospect there probably should be a difference between passive and object-relative RCs, the current semantic encoding makes no distinction between a passive RC with a by-phrase, "*who was bitten by the dog*," and an object-relative, "*who the dog bit*." This will cause problems for the production network and there probably should be a distinction in emphasis between these two clauses.

The PP bit is used in conjunction with an object modifier when that modifier is expressed using a prepositional phrase. This may seem like a syntactic distinction, rather than a semantic one. However, some small distinction in semantics is necessary if the sentence production system is to know that it should produce a double object dative or a prepositional dative. One might think of this as another matter of emphasis.

Object features

With a large vocabulary, it would be most efficient to encode the semantic properties of objects using a set of shared features, like LIVING, MOSTLY-HARMLESS, and MADE-OF-FOAM. If individual features do not denote particular objects, a large set of objects can be represented with a distributed pattern over a much smaller set of features. This has the further advantage of conveying the relative similarity of the items. However, with a small vocabulary, the advantage of using shared features disappears because many of the features will refer only to a single object. If HAS-TAIL, SHAGGY and BAD-BREATHED are only used for *dogs*, it would be more efficient to simply replace them with a DOG bit.

Thus, because Penglish has a fairly small vocabulary, it represents objects and actions using semi-distributed encodings involving a mixture of shared and unique features. Thirty-four bits are used in a distributed fashion to encode properties that might be shared by more than one object. These include general categories like OBJECT (*knife*), PLACE (*park*), LIVING (*dog, boy*), and ABSTRACT (*story*). Most objects use just one of these features, but a *book*, for example, is both an OBJECT and ABSTRACT. Other shared bits, like VEHICLE, HUMAN, and FOOD, encode subcategories of those objects. Still others, like LONG, OLD, and HARD denote salient features of those objects.

Some objects are encoded entirely using shared features. A *boy*, for example, is LIVING HUMAN SMALL YOUNG MALE. The encodings of most objects, however, involve unique identification bits, like LAWYER, BIRD, and TRUMPET. The objects in Penglish use an average of 4.5 active bits in their encodings.

An additional set of features is used to represent properties of specific instances of a type of object. The PLURAL bit, for example, indicates when there is more than one of the item. A better semantic representation should probably identify the number of an item at some finer grain, like ONE, TWO, THREE, A-BUNCH, and A-WHOLE-LOT. Other bits encode the definiteness of the object, whether it is definite, indefinite, demonstratively specified with THIS or THAT or quantified with SOME.

Action features

Like objects, actions are encoded using semi-distributed representations. There are four main classes of action: ACTION, MENTAL, SENSE, and STATE. ACTION is used for physical acts, MENTAL for verbs like *think* and *believe*, SENSE for verbs like *see* and *feel*, and STATE for *have*, *involve*, and *consist*. Some verbs use more than one of these types. *Ask*, for example, is both an ACTION and MENTAL and *read* is both a SENSE and MENTAL.

There are also additional shared modifiers that contribute to an action's meaning, like GIVING, GETTING, UNWILLED, SPOKEN, or AFFECTING. As with the objects, most actions then have a unique identifier, like BITE or EXAMINE. Actions in Penglish have an average of 3 bits in their base representation.

Finally, a few more bits encode the tense of a specific occurrence of an action, PAST, PRESENT, or FUTURE, and its aspect, SIMPLE, CONTINUOUS or PERFECT. Specifying these latter distinctions as such may be a bit too specific to the encoding of English semantics as opposed to the semantics of other languages.

Property features

Finally, the properties of objects and actions, specified with adjectives and adverbs, have their own featural encoding. They all share the PROPERTY bit, which is distinct from the PROPERTY bit used in the relation because that only occurs in the middle element of the proposition. Two other shared bits are BAD and GOOD, which are used with adjectives or adverbs that seem to be consistently judgmental. Otherwise, most adjectives and adverbs are encoded with unique bits, like LOUD or YOUNG. Note that *hard* can mean either DIFFICULT or NOT-SOFT and *nice* can mean either KIND or CLEAN. Note, too, that many of these same property bits are used in the encoding of objects that tend to have those properties. For example, BIG, OLD and MEAN are part of the meaning of the word *lawyer*. But that does not prevent some lawyers from being young or, even, nice.

The temporal adverbs *yesterday* and *today* have a special encoding. They use the WHEN bit along with either the PAST or FUTURE property used for actions.

5.5.5 Syntax to semantics

Sentences in the Penglish language are not generated semantics-first, but are produced using a syntactic grammar. One problem, then, is how, given a sentence, to obtain its true propositional meaning for the purpose of training and testing the connectionist model. This would seem to require another whole comprehension system. Fortunately, this problem is not so difficult because, in determining a sentence's message, we have more to go on than its surface form. When a sentence is generated using SLG, we can also obtain its syntactic structure along with its surface form. Translating from a surface form to a propositional message turns out to be fairly easy in Penglish.

A script was written to process the syntactic tree of a Penglish sentence and generate the appropriate propositions.

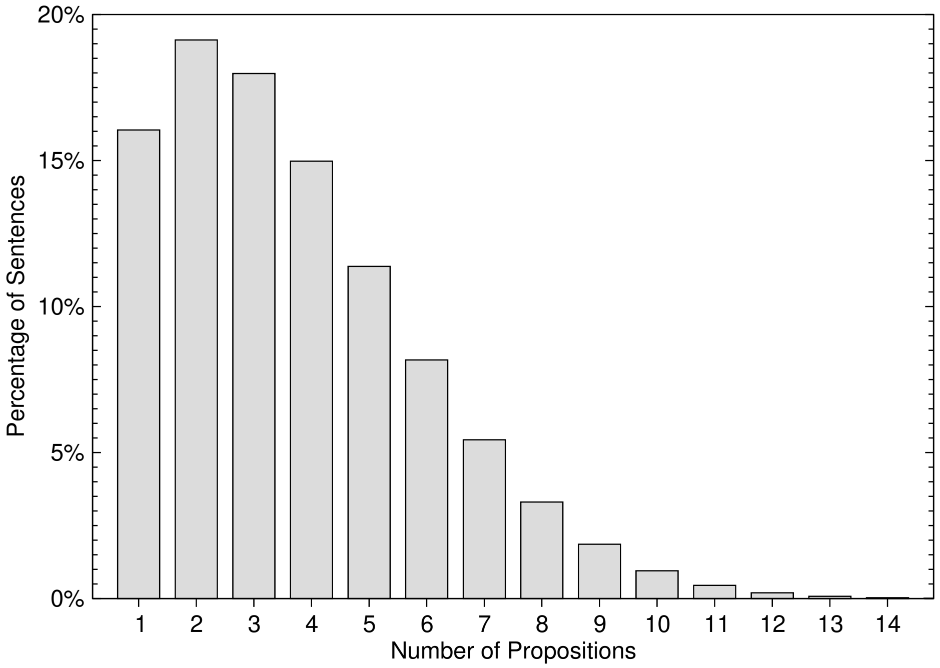


Figure 5.3: The distribution of the number of propositions per sentence.

The script, essentially, recursively transverses the parse tree and outputs propositions as soon as their major elements can be identified. The order of the propositions, then, will convey some information about the syntactic structure. However, that information is likely to be redundant with information contained in the propositions themselves. The order of the propositions was not carefully controlled as it is not expected to have a major effect on the model. In retrospect, this is an issue that deserves further study. One consistent aspect of the proposition ordering is that the first proposition tends to encode the subject of the sentence. In this way, one could think of the subject as emphasized, although this is redundant with the fact that the EMPHASIS bit is used in conjunction with the thematic role of the subject.

5.6 Statistics

The grammar used to describe the Penglish language is, in complexity, nearing the limit of what can currently be handled by the SLG program. It takes about a day on a 500 MHz 21264 Alpha processor to convert the grammar from SLG form into stochastic CFG form. Counting different morphological variants of the same root, Penglish has a total of 315 words. Its grammar, after compilation into a SCFG, has 13,534 non-terminal symbols which use 1,285,495 transitions, or productions. There are 94.98 average transitions per non-terminal and 2.04 average symbols per transition. The file encoding the grammar, in reduced binary format, is 26 megabytes long, compressible to 14 megabytes. When converted into Chomsky normal form for parsing purposes, the grammar has 78,339 nonterminals using 1,250,300 transitions, with an average of 1.99 symbols per transition.

Figure 5.3 shows the distribution of the number of propositions per sentence. A simple intransitive would have one proposition and a simple transitive would have two. Most basic single-relative clause sentences have four propositions. Because sentences with more than eight propositions were not of much relevance to any of the planned experiments, and because it was not expected that the network would be able to handle sentences of greater complexity, only those with eight or fewer propositions were used in training the network. This required filtering out 3.6% of the sentences.

Figure 5.4 shows the distribution of the number of words per sentence in Penglish. This does not count the period at the end of a sentence or the initial *begin* symbol prior to the first word in the sentence. Limiting the semantic complexity of the sentences to less than or equal to eight propositions only causes a slight skewing in the overall

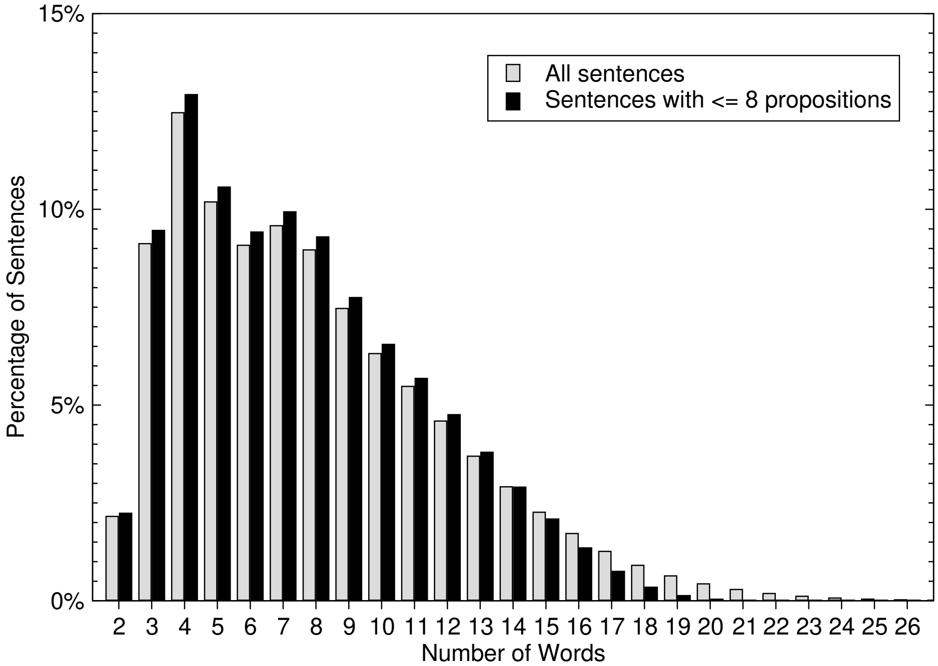


Figure 5.4: The distribution of the number of words per sentence.

distribution of sentence lengths.

Chapter 6

The CSCP Model

The immediate goal of the connectionist sentence comprehension and production (CSCP) model is to learn to comprehend and produce sentences in the Penglish language, a simplified form of English that was introduced in the previous chapter. The broader goal is that, in doing so, the model will account for a wide range of human sentence processing behaviors. This chapter describes the CSCP model—its architecture, its operation, and how it is trained and tested. Section 6.1 describes general aspects of the units and learning mechanism employed in the model. Section 6.2 explains the portion of the model responsible for encoding sentence meanings from a set of propositions to a static, fixed dimensionality message representation and decoding them back into propositions. Section 6.3 discusses the heart of the model—comprehension, prediction, and production. Sections 6.4 and 6.5 explain how the model was trained and tested, including the measures used to assess comprehension and reading time and the theory behind them. Finally, Section 6.6 clarifies the critical claims of the model as well as acknowledging some of its major limitations.

6.1 Basic architecture

The CSCP model operates as a backpropagation-through-time, simple-recurrent network (Rumelhart et al., 1986; Elman, 1990). The term *simple-recurrent network* (SRN) has somewhat different meanings for different researchers, but I take the difference between simple-recurrent and fully recurrent networks to be primarily a matter of the manner in which information propagates through them. In an SRN, as in a feed-forward network, information flows all the way through the network, from the input to the output, in a single step. An SRN network with one hidden layer will update the input layer, update the hidden layer based on the new input representation, and then update the output layer based on input from the new hidden representations, all in a single step. Thus, in one step information can flow through all layers in the network.

In a fully-recurrent network (FRN) (Pineda, 1989; Pearlmutter, 1989), on the other hand, unit updates are synchronous throughout the network and consist of two phases. First, all units compute their inputs based on the current outputs of other units. Then all units update their outputs based on the current input. Thus, in a single step of an FRN, information will flow over just one set of links. As a result, FRNs must normally be run for many activation update steps following a new input for the information in the input to reach all parts of the network. An SRN, on the other hand, will often have just one activation update step following each input.

Another difference between SRNs and FRNs tends to be in the behavior of the units themselves. With FRNs, it is standard to use units with hysteresis, or time-averaging, of their inputs or outputs. An output-averaged unit will compute its output, or activation, as a weighted sum of its previous output and the output that would otherwise result from its current input. It is also possible to have units with time-averaging in their inputs, prior to any non-linear transformation from input to output. In either case, the units will tend to respond slowly to any change in input. On the other hand, the current model, like most SRNs, uses units whose outputs are based entirely on their current input, with no influence of previous outputs.

There are a number of advantages to using an FRN with time-averaged units, particularly in modeling cognitive processes. Computation in an FRN is a smooth, dynamic process. The network cannot easily jump from one state to a vastly different state—it must undergo a gradual progression through state space. This is presumably a better model of the dynamics of real neural systems. As a result of this gradual, highly interactive process, FRNs can develop *attractor-state* dynamics (Hopfield, 1982). Attractors are activation states of the network that are stable over time and are not disturbed by small perturbations. If the external input to the network remains constant, most such networks will gradually settle into an attractor state. A well-trained network will tend to develop attractors that are very close to the desired output on a given input. If the network is in a state that is close to an attractor, it will tend to be drawn in to the attractor. This enables the network to "clean up" faulty representations, potentially making it more tolerant to noise or damage.

Another important feature of FRNs is that processing time in the network can be a direct model of human processing time on a similar task. Some inputs will result in faster settling than other inputs, either because they are easier to process and have a stronger attractor or because the network begins at a point in state space that is closer to the eventual attractor. FRNs with attractor dynamics have been the basis for several successful cognitive models, particularly in the domains of associative memory and of word reading (Hinton & Shallice, 1991; Plaut & Shallice, 1993).

The primary disadvantage of using an FRN is the practical problem of the time it takes to simulate one on a serial computer. To process a single input, an FRN will often take 10 to 50 times as long as an SRN because of the length of the settling phase. In models for which attractor dynamics are not expected to be of critical importance, using an SRN is often a better choice. In the same period of time, a much larger SRN can be trained, enabling the model to handle problems of greater complexity. For this reason, an SRN was used for the CSCP model, rather than an FRN. However, this should not be taken as a theoretical commitment, it is merely a matter of expediency. Given the available computational resources, using an FRN might have meant a training time of three years, rather than two months. But it is likely that an FRN with attractor dynamics might be a better model of human sentence processing, and the current model may act inappropriately in some circumstances because it is not fully recurrent.

The use of an SRN does raise one important question, which is, how to obtain a measure of processing time, or, more specifically, reading time. In an SRN that is constrained to process each input in a single step, there is no direct correlate of time. Therefore, we must invent a surrogate for settling time based on other measurable properties of the model's behavior. Sections 6.5.2–6.5.4 discuss how that was done in the CSCP model.

Following each step of the forward phase, the network's output layer, or layers, reaches a certain pattern of activation. Normally, a target activation pattern is also provided by the environment, specifying the desired output of the network. The difference between the actual and target outputs is the network's *error*. How this error is computed for the CSCP model will be explained in the next two sections. When training a neural network, we seek to make changes in the link weights that are proportional in magnitude (although opposite in sign) to the partial derivative of the error function with respect to the link weight. Computing these partial derivatives is the role of the backpropagation phase.

One major difference between the CSCP model and more traditional SRNs is in the way error derivatives are backpropagated through the network. Standard SRNs have a backpropagation pass immediate after each forward pass (Elman, 1990). The network's units are updated in the forward pass, error is computed, and the error derivatives propagate back through the network in the backward pass before the next forward pass occurs. In a standard SRN, the error derivatives used to train the recurrent connections only reflect the error that occurred as a result of the immediate outputs of the network. As a result, the recurrent connections are only trained to retain information in the network's current activation state for the purposes of producing appropriate outputs on the very next step. There is, however, no consideration of whether that information may be useful at some later point in time. Despite this limitation, SRNs are still capable of learning tasks that require information to be stored for many time steps before it is used (Rohde & Plaut, 1999). However, such networks will have difficulty learning a task that requires information to be retained over long periods of time during which that information is of no immediate utility (Bengio, Simard, & Frasconi, 1994).

A more powerful method for learning temporal tasks is backpropagation-through-time (BPTT) (Rumelhart et al., 1986). While a simple-recurrent network interleaves forward passes and backward passes, one for each time step, a BPTT network runs all of its forward passes first and then performs all of the backward passes. Essentially, this is equivalent to unrolling the network in time—making a duplicate copy of it for each time step. In this unrolled network, recurrent connections project from each sub-network, or copy of the network for a particular time step, to the sub-network representing the next time step. Information propagates through this larger network, through time essentially, in a single forward pass and error is calculated for each of the sub-networks. Then there is a single backpropagation phase that starts at the output layer on the last time step and runs back through all of the sub-networks to the input layer on the first time step. In doing so, for each link in the unrolled network, the procedure calculates the derivative of all future error (on the current example) with respect to the weight of the link. The error derivative for a link in the original, pre-unrolled, network is the sum of the error derivatives for all copies of the link in the unrolled network.

The advantage of using BPTT is that it is a more powerful method for learning to process temporal sequences, but requires no more computation than the normal SRN method. A major disadvantage, however, is that it is not a reasonable approximation to any biologically plausible mechanism. BPTT requires that activation and error information be stored for all units in the network for each time step and that activation states be re-experienced in reverse order. BPTT was used in the CSCP model for expediency. The final behavior of the model is of principal interest, as well as the overall progression of its learning, but the exact mechanism by which the model learns, is of less importance in the current study. I do not expect that a true SRN model would have substantially different behavior, qualitatively, but that remains to be seen.

The architecture of the complete CSCP model is shown in Figure 6.1. The value in the lower-left corner of a group indicates the number of units in the group. Solid arrows represent complete projections, with links from each unit in the sending group to all units in the receiving group. Although not shown, all non-input units also receive a trainable input from a *bias* unit. The size of the depicted groups is suggestive of, but not exactly proportional to the number of units in the group. All self-recurrent connections, as well as the connection from the message layer to the comprehension gestalt layer, are delayed lines. The input to these links is drawn from the output of the sending layer on the previous time step. Such a connection in an SRN is often depicted as a delayed copy of activations to a *context* layer, followed by a standard projection back to the original layer, and that is, in fact, how it is implemented in LENS. However, depicting the context groups here would make for a very confusing figure.

6.2 The semantic system

In order to perform sentence comprehension and production, the model must have the ability to work with sentence meaning, including providing messages as a source for production and testing the messages that result from comprehension. The CSCP model is composed of two separable parts, the comprehension/production system and the message encoding/decoding system. We will start with an explanation of the latter.

6.2.1 Slots vs. queries

As discussed in Section 5.5, the meaning of a Penglish sentence comprises a set of propositions. One approach to representing sentence meaning might be simply to have a bank of groups, or slots, each one storing a single proposition. If a sentence uses four propositions, then only the first four slots would be used. Obviously, some limit would have to be placed on the maximum number of propositions per sentence, which in itself is somewhat questionable. But this approach to message representation has some redeeming features, the principal one being that it is simple and straightforward. The bank of propositions can directly serve as the source for production and the output of comprehension.

However, the proposition-bank approach has some serious drawbacks as well. In general, slot-filler representations that use physically separate, parallel slots tend to hinder generalization. There is often considerable mutual information in the representations used in different slots. The same propositions, or types of propositions, might occur in all different locations. If the slots are physically separate, any information that is useful in processing those propositions must be learned and represented redundantly. Either that, or the information must be represented at a deeper level in the network where it can be used in all slots, which creates additional problems. As a result, slot-filler representations inhibit generalization. Knowledge learned in processing one slot is not easily transferred to the processing of other slots. This may not be so noticeable for the slots that are used frequently, but the model will be at a loss to do anything with the slots that are used only in rare, complex sentences, and performance will drop off very quickly with sentence complexity.

Another problem with the slot-filler model is that it adds an extra dimension to comprehension. Not only must the network produce the right set of propositions, it must produce them in the correct positions. This may be difficult because it requires global organization and because, in the current design, the ordering of propositions conveys very little information. The position of one proposition will depend on other aspects of the sentence and, possibly, on information that arrives after the proposition is ready to be stored. This may mean that propositions are moved around during comprehension. In doing so, the model must be sure that the propositions remain intact and that one part is not

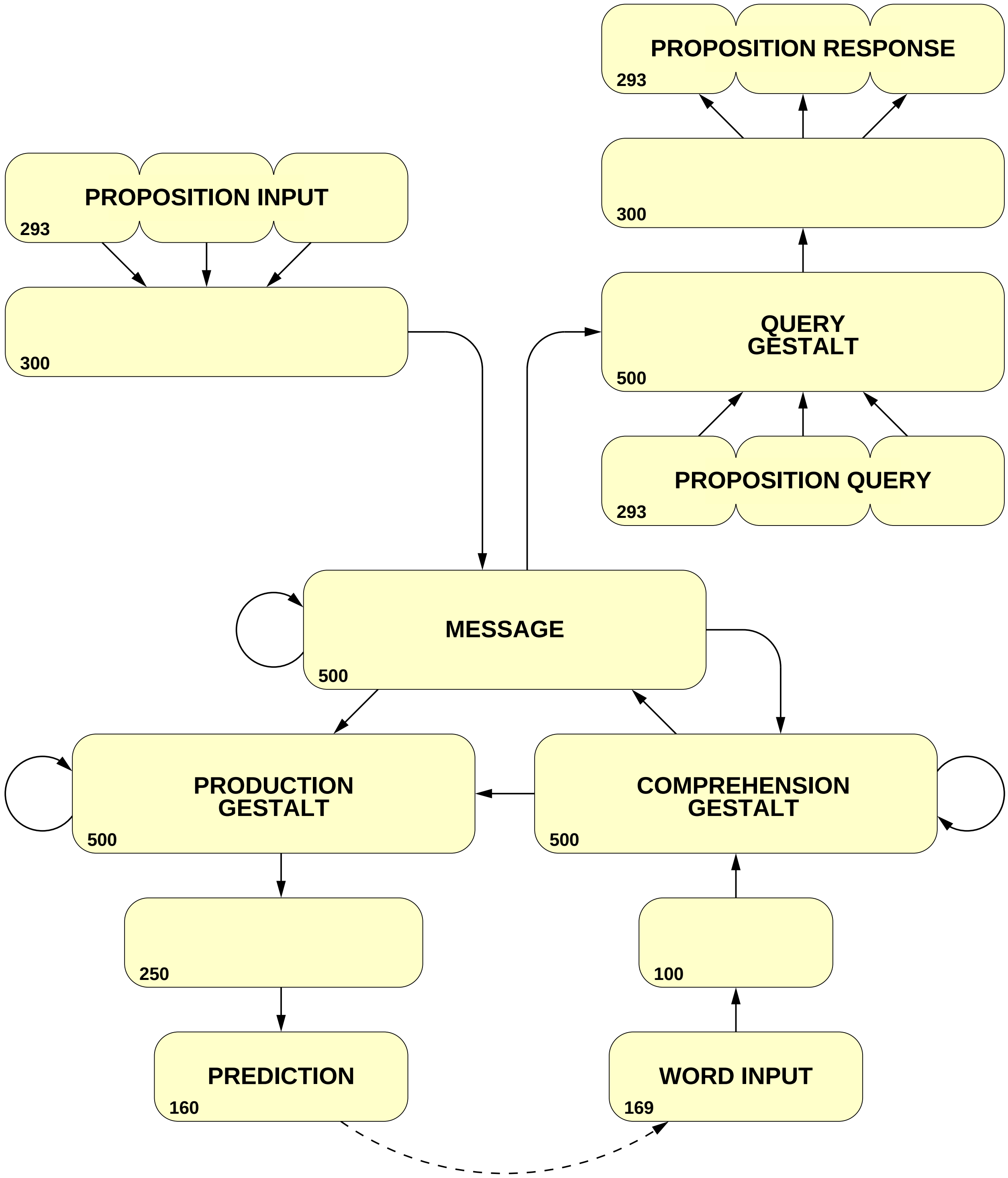


Figure 6.1: The full CSCP model.

left behind. If such a network makes a mistake, like switching the objects that play the roles of agent and patient, it will not be clear if the problem is one of comprehension difficulty per se, or a failure to align propositions with slots.

Therefore, the simple slot-filler representation of meaning may be a poor choice for a connectionist model of complex sentence processing. An alternative is suggested by the work of St. John and McClelland (1988, 1990, 1992) (see Section 2.2). They devised a model of comprehension whose performance was evaluated using a *querying* mechanism. After "listening" to a sentence, the model was asked questions, in the form of probes, to which it responded with an answer. The model could be probed with a thematic role (e.g., agent), to which the correct response is the object (noun) that fills that role, or the model could be probed with an object to which the correct response is the role played by the object. Essentially, then, the model is using the same set of units for all propositions and only explicitly represents a limited amount of its knowledge at any one time.

In order to handle the propositions used in the Penglish language, a somewhat more complex type of probe is required. It is not sufficient to probe the network with just a thematic role, like agent, because there may be several verbs in the sentence, each with its own agent. Recall that propositions used in the Penglish semantics involve three parts. In the case of a thematic role, the first part will be the action (verb), the second part the role, and the third part the object (noun). A probe in the CSCP model consists of a proposition with one of these parts missing. The correct response is the complete proposition with the missing part filled in. Thus, one can ask three questions about a proposition by leaving out any of the three parts.

There is some potential for ambiguity in this form of querying. A sentence may have an object or action repeated, as in, "*The dog chased the cat and the cat chased the mouse*." In this case, it is not clear who the agent of *chased* is or what role the *cat* should play with the verb *chased*, since there are two possibilities for each. Ultimately, this potential ambiguity is a significant limitation of the current representation. In practice, the problem is not too serious because any distinguishing characteristics, including number, definiteness, and tense, will serve to disambiguate two objects or two actions. Thus "*The dog chased the cat and the cat will chase the mouse*," is not problematic because the actions differ in tense.

Ambiguous queries turn out to be fairly rare in Penglish, and the problem they create can be mitigated by simply avoiding any questions that are ambiguous. Overall, about 0.86% of queries in the testing set are ambiguous. Figure 6.2 shows the percentage of ambiguous queries as a function of the number of propositions in the sentence. Even with 8 propositions, they account for less than 1.5% of all queries. The relatively high percentage of ambiguous queries for sentences with two propositions turns out to be due to reflexive transitives involving *I/me*, *you*, or *something*, which do not use determiners. Penglish does not include *myself*, *yourself*, or *itself*, which would be used in English in such cases.

6.2.2 Message encoding and decoding

A direct query mechanism is sufficient to train a comprehension-only network, like the St. John and McClelland model, mentioned above and in Section 2.2. The output of the comprehension system can serve as a static input to the query system, and error derived from imperfect responses can be backpropagated through the network to train comprehension. However, such a mechanism is not sufficient as a basis for production. One must be able to give the production network the full meaning of the sentence, or message, in advance of the production process. Therefore, it is not sufficient to be able to decode messages—it must also be possible to encode them, from a set of propositions to a static message.

Message encoding in the CSCP model is a relatively simple process, carried out by the left half of the encoder/decoder system, depicted in Figure 6.3. Again, using a parallel slot-filler input to the message encoder may result in poor generalization and information sharing. Therefore, the message encoder processes the propositions sequentially, using a single set of input groups.

The encoder network, like nearly all of the CSCP model, uses logistic, or sigmoidal, units with a temperature of 1.0, which is the same as a gain of 1.0. Activations in such units range from 0 to 1. Prior to encoding, the activations of all units are reset to 0.5. The propositions are then loaded sequentially, one proposition per time step. Each proposition is loaded by setting the activations of the proposition input units to the vector representation of the proposition (see Section 5.5.4). Activation then flows to a hidden layer, the purpose of which is to allow the network to perform some initial processing and to recode the proposition into a more useful form. This layer then projects to the final step of

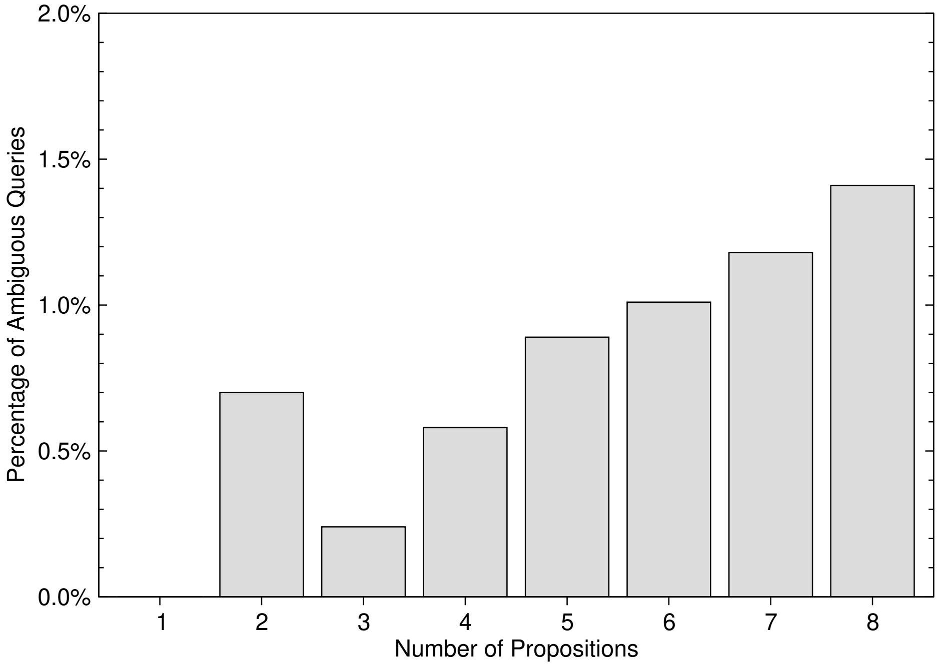


Figure 6.2: The percentage of ambiguous queries as a function of the number of propositions per sentence.

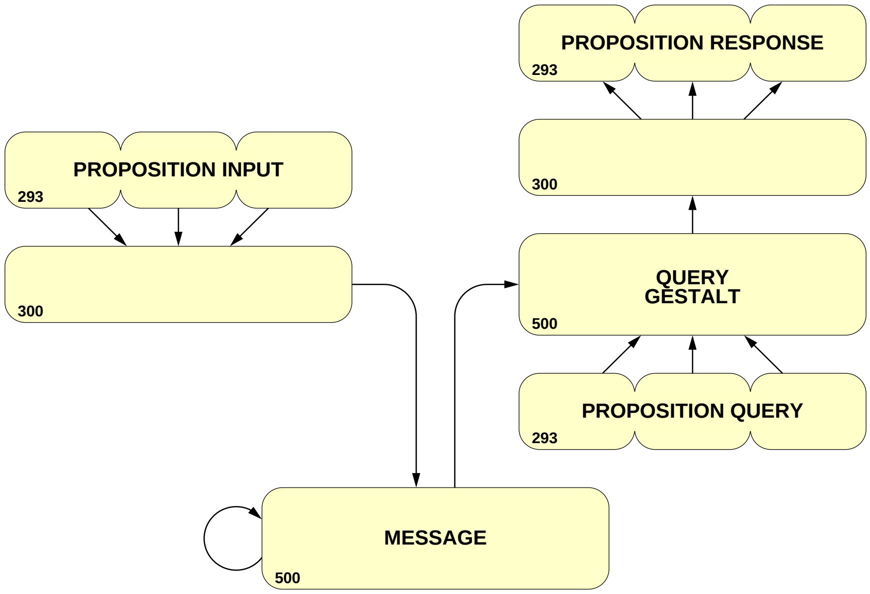


Figure 6.3: The message encoder and decoder system.

the encoding process, the message layer.

The message layer is responsible for storing all of the propositions seen so far for the current sentence, which it does by virtue of its recurrent projection. Essentially, the message layer is a memory. As each proposition comes in, the message layer must combine the new information with its previous representation to form a new one that retains all of the important information. Because each proposition uses 293 units and the message layer has just 500 units, it will not be able to store more than one proposition unless it *compresses* the information. The more propositions that must be stored, the more they must be compressed.

But encoding messages is just half of the problem. In order to train and test the encoding process, messages must also be decodable into their constituent propositions. The right half of the network in Figure 6.3 is responsible for decoding messages. As mentioned earlier, a message is decoded by presenting a probe consisting of a proposition with one part of it left blank. The correct response is the complete proposition, with the missing part filled in.

As one might expect, partial propositions are entered at the proposition query input layer. This projects to the query gestalt layer, which also receives input from the message layer. The query gestalt must combine the information from the query and message to produce the complete response. This passes through a hidden layer before finally reaching the proposition response output layer. The cross entropy error measure is used to assess the disagreement between the model's output and the correct response. Cross entropy is defined as follows, where o_i is the actual output of unit i, and t_i is its target output:

$$\text{cross entropy} = \sum_i t_i \ln\left(\frac{t_i}{o_i}\right) + (1 - t_i)\ln\left(\frac{1 - t_i}{1 - o_i}\right)$$

In addition to cross entropy, a *zero-error radius* of 0.1 was used at the proposition response layer. Essentially, if the output of a unit is within 0.1 of the correct value, no error is assessed. Using a zero-error radius in a situation like this, where the targets are binary, encourages the network to focus on the units that are really wrong. Otherwise, much of the weight changes will be targeted toward making small improvements in units that are already close enough to the correct value, ignoring the one or two units that are really off the mark.

The ability to encode sentence meanings into a static representation was necessary to model production, but it will also lead us to reconsider how the comprehension system is trained. In the St. John and McClelland (1988, 1992) model, the product of comprehension was directly queried to provide it with a training signal. After each word in the sentence, the network was probed with all possible questions to see how much of the full meaning of the sentence had been extracted. Thus, the model was encouraged to predict as much of the message as was possible given the available input. The problem with this approach is that it is computationally expensive. If applied to the current model, the total number of probes would be three times the product of the number of propositions and the number of words in the sentence. For a typical sentence with six propositions, that means about 216 probes.

Aside from the inefficiency, a more serious problem with directly querying the product of comprehension is that the message produced by the comprehension system might bear little resemblance to the message used as input to the production system. The similarity in representation between messages used for comprehension and production is an important aspect of the current model. Therefore, rather than querying the product of comprehension, the CSCP model uses the encoded message itself as a target, providing direct feedback to the output of the comprehension system. Therefore, both comprehension and production share the same message representation. The comprehension and production system is explained further in Section 6.3.

6.2.3 Training the semantic system

The semantic system is actually trained separate from and prior to the rest of the model. Feedback used in training the encoder portion of the system derives from error resulting from querying the decoder part. Thus, the encoder and decoder interact in developing the representations of sentence meaning used in the message layer. The training process operates by loading propositions one by one and, after each proposition is loaded, asking the decoder to answer all possible questions concerning the propositions loaded thus far.

For example, take a sentence whose meaning involves three propositions. The first proposition is loaded in the proposition input and the activation flows through to the message layer. This is followed by a querying phase. The output of the message layer is held fixed while the three probes are presented in the proposition query layer, resulting in outputs at the proposition response layer that generate an error signal. This error signal would then be backpropagated, but let us ignore that for a minute. In the next forward time step, the second proposition is loaded, updating the message, which should now contain the information from the first two propositions. The message layer's activation is again frozen and both the first and second propositions are queried, totalling six questions. Finally the third proposition is loaded and the first, second and third propositions are all queried. For a sentence with n propositions, this procedure will perform $3n(n+1)/2$ queries. Thus, the training time on a sentence is quadratic in the number of propositions.

Now we return to the issue of backpropagation. After each query, error is assessed at the proposition response layer. It is immediately backpropagated through to the proposition query layer resulting in the accumulation of an error derivative for each link. Error is also backpropagated to the message units, but not beyond them. During a querying phase, the derivatives of the error with respect to the outputs of the message units accumulate. At the end of the query phase, those error derivatives are stored. Once the whole sentence has been loaded and queried there is a single backpropagation through time involving the message layer, the proposition input layer and the hidden layer between them. At each step in this process, the message layer error derivatives that were stored during the corresponding querying phase are injected back into the message layer. This is equivalent to a standard backpropagation through time, except that the error derivatives on the message layer at each step do not derive from immediate targets but are the accumulated product of a number of queries.

In summary, the decoder part of the system is trained using standard single-step backpropagation, while the encoder part of the network is trained using backpropagation through time. The reason for using BPTT, rather than standard SRN backpropagation as in St. John and McClelland (1988), is simply that it achieves better results in a task such as this because the error signal provides direct feedback on the future utility of information.

The encoder/decoder network is quite large, comprising 1.1 million links, and training on a sentence can involve a number of steps. As a result, training time is a serious issue and a straightforward implementation would be prohibitively slow. Most of the running time in this system is due to the decoder, especially on complex sentences, and certain shortcuts can be used to improve the speed of the decoder.

In any large network, the vast majority of the time required for a forward pass is taken up by computing the inputs to the units, because each input involves one or more dot-products of moderately large vectors. The worst point in the decoder model is in computing the query gestalt input. Each query gestalt unit is receiving 500 inputs from the message layer and 293 inputs from the proposition query layer. However, throughout training on a sentence there is a lot of duplicate processing going on. During a querying phase, the activations of the message layer do not change. Therefore, the contribution of the message layer to the input of a query gestalt unit does not change during this period. Furthermore, the same probes are fed to the network repeatedly. Questions about the first proposition are asked in each of the query phases.

Therefore, the contribution of each message to the query gestalt input and the contribution of each probe to the query gestalt input is stored. The links from message to query gestalt are only traversed once for each message and the links from proposition query to query gestalt are only traversed once for each probe. To compute the actual input to a query gestalt unit, the simulator need only recall the appropriate contributions from the message and query gestalt layers and add them up, along with the contribution from the bias unit. Similar shortcuts are used in the backward phases. Rather than backpropagating across these connections immediately, the derivatives of the error with respect to the query gestalt layers are stored and accumulated and then backpropagated only once for each different input. Although these optimizations make the implementation of the semantic system rather complex, they do significantly improve its running time.

6.2.4 Properties of the semantic system

The semantic system is basically a sequential auto-associator. A traditional form of auto-associator is a multi-layer network that maps an input to an identical output via a bottleneck that forces the network to compress the information in its input. The semantic system used here is similar, but has been extended to handle sequential inputs. Because this network must learn to form its own message representations and, in so doing, is forced to develop representations that compress the propositional information in the sentences, messages that are easier to compress will, presumably, be easier to recall. It is hypothesized that the performance of this semantic system will primarily be influenced by two factors: *frequency* and *confusability*.

Sensitivity to frequency

All effective compression methods operate by taking advantage of frequency. If more frequent items are assigned smaller representations, the overall code will be more efficient. For example, in the well-known Huffman encoding, the number of bits assigned to each symbol is monotonic in the inverse frequency of that symbol (Huffman, 1952). Thus, we should expect that the semantic system, as a compressor, will principally take advantage of frequency in its message encoding.

However, the parallel with information-theoretic compression methods is not precise. Such signal processing models assume that encoding takes place over a sequence of non-decomposable symbols. But the propositions that serve as inputs to the encoder are composed of distinct elements and those elements are composed of individual bits with various co-occurrence patterns. All humans and all animals are living. Young things tend to be small. Physical actions tend to have animate agents, while mental actions tend to have human agents. Each of these regularities, although they are not strict rules, are properties that the encoder can leverage in compressing the propositions. The types of regularities that the system can take advantage of can exist on many levels. The system may be sensitive to the frequency of a given action, object, property, or relationship overall, or the frequency of a given object in a given thematic role, or the frequency of an object in the agent, patient, or other role for a given verb, or the frequency of an object having a certain property, or the frequency of the agent of a particular verb having a certain property. The list goes on.

To make this a bit more concrete, consider an example. Dogs, in Pengland, often bite. Therefore, the encoder should have little trouble storing the proposition [bite, agent, dog]. Girls are less likely to bite, so [bite, agent, girl] will require more representational capacity and may interfere with other propositions, but shouldn't be too difficult because girls are often agent of other verbs. Furthermore girls share an important feature with other things that often bite—they are all living. On the other hand, [bite, agent, ball] would be semantically anomalous and would thus be harder to store. But balls do appear as agents of other verbs, so the encoder may still have the ability to store and retrieve such a proposition. To take it a step further, however, [bite, agent, sleep] would be completely invalid and the semantic system would probably fail entirely to encode such a proposition.

When the encoder becomes overburdened, its messages may become underspecified or incoherent. In this case, the decoder will do its best to answer the queries based on its own experience, and in so doing, it is expected to make errors that tend to reflect regression to more frequent or more likely possibilities. A cat following a car might turn into a dog following a car. A reporter who is supposed to be the recipient of a book might become the author of the book. An old lawyer might become a mean lawyer.

Sensitivity to confusability

The other main factor that the semantic system is expected to be sensitive to is *confusability*. This is not a matter of the overall frequency of individual propositions, but of interactions between propositions within a sentence. We have already mentioned the potential problem of ambiguous queries. If two propositions overlap in two of their parts, but differ in the third, and the model is asked to fill in this third part, there are two possible answers and it can be correct, at best, only half of the time. This is an extreme form of the confusability problem, but other pairs of propositions can result in confusion without reaching the point of complete ambiguity provided they nearly overlap in two of their parts.

Consider the examples below. In (45a), there are mean dogs, [dogs, property, mean], and there is a nice dog, [dog, property, nice]. The representations for *the dog* and *the dogs* are quite similar and differ only in number. Therefore, answering the question [dog, property, ?] will be fairly difficult, because it requires the network to rely on the single bit that encodes number.

(45a) The mean dogs bit the nice dog.

(45b) The boy followed the girl who was following the dog.

(45c) The boy who was following the dog followed the girl.

(45d) The boy forgot the girl who was following the dog.

In (45b), the *girl* is both the patient and agent of following, although the two following acts are distinguishable because they differ in tense. Therefore, the question [followed, ?, girl] will be a difficult one. Similarly, [followed, agent, ?] is also hard, because both the *boy* and *girl* were following and tense is again critical. The question [?, agent, girl] may also be difficult because the network is storing two similar actions and it must produce the one with the correct tense. Recalling that the action involves following will not be difficult, but getting the tense correct will be.

In contrast, a center-embedded subject-extracted relative clause, as in (45c), results in a slightly different set of confusions. In this case, it is still difficult to fill in an action given an object or a patient given a verb, but it is easy to respond to the question [following, agent, ?] because the decoder need not pay attention to tense to answer the question right—the boy is the agent of both actions.

Most relative clause sentences, however, will not involve duplicate actions, but they still may result in confusion. Sentence (45d) is identical to (45b), but one of the verbs has changed. In this case, the question [following, ?, girl] may still be difficult because there is interference due to the fact that the girl is the agent of one action and the theme of another. The degree of interference may depend on the degree of similarity between the actions. This property of the semantic system is very similar to the problem of *perspective shift*, cited by MacWhinney as a factor in the difficulty of certain types of relative clauses (MacWhinney & Pléh, 1988), but it is here generalized beyond just changes in subject.

That the semantic system is sensitive to frequency and confusability is, at this point, merely a prediction. I have not yet provided any evidence of this. However, given our understanding of other connectionist association models, it was the expectation that the system would behave in this way that led to this choice of design for the semantic encoder and decoder.

The fact that the semantic system has trouble recalling rare or confusing propositions may be seen as a limitation. But it is a perfectly reasonable limitation that may very well reflect similar limitations of the human memory systems in general, and semantic memory in particular. The semantic system should not be viewed as extraneous to the CSCP model. It is an integral part of the complete language processing system, and it, in addition to those parts of the system involved in syntactic operations, will be responsible for much of the behavior evidenced by the model.

6.2.5 Variants

A number of decisions went into the design of the semantic system and alternative formulations are possible that may have somewhat different properties. One such decision is the fact that the model reproduces the entire proposition in response to a query. An alternative is to respond just with the missing component of the probed proposition and to forget about the given components. A smaller proposition response group could then be used to encode any of the three parts of the proposition. The proposition-completion approach was used instead because it was thought that having a single group used for different parts of the proposition might be confusing. If the network knew the answer, but forgot the question (which part was supposed to be filled in), its response might be incoherent.

One addition that might be useful is an extra bit for each of the three parts of the proposition query group. One of these bits would be activated to signal which part of the probe had been left blank. Currently, the network must detect the empty component of the probe based on the overall pattern of activation. It is possible that adding indicator bits might result in a small improvement in performance.

Another potentially useful addition is some extra information provided at the proposition input. If the model were told in advance how many total propositions were on their way, it might be able to adapt the level of compression. If it knows that only three propositions are coming, the first few can be compressed only slightly. If it knows that eight propositions are coming, it could start with a higher level of compression and avoid a lot of later reorganization. It is possible that this could lead to better performance, but it also may have some effect on the way in which messages are encoded that could be either helpful or harmful to the comprehension system. Neverthless, this seems worth investigating at some point.

One final issue is the frequency with which different propositions are queried during training. The first proposition is queried after every encoding step. The second proposition is queried after all but the first encoding step. The last proposition to be loaded is only queried once. This is by no means the only reasonable strategy. It may very well have the property that the early propositions will be learned better than the later ones. An alternative might be to load

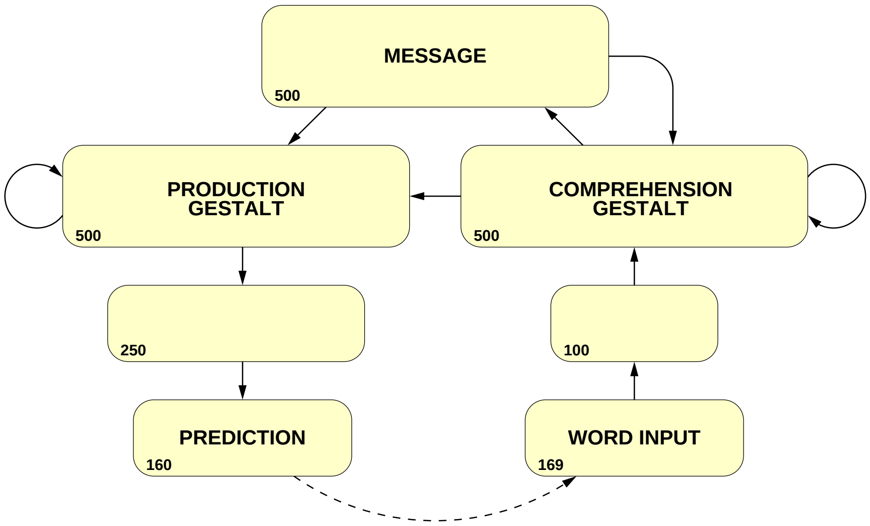


Figure 6.4: The comprehension, prediction, and production system.

all of the propositions and then query all of them once. But this would seem to bias the recall in favor of the later propositions, because the encoding of the first proposition is far removed from the error signal. It is possible for these options to be blended by querying propositions probabilistically. This may be another area for future work.

Nevertheless, it is expected that the current training method will result in U-shaped performance. Late propositions will be recalled well because they are more recent and early propositions will be recalled well because they are more often queried, and possibly because they receive extra representational space in the semantic encoding because they were stored first. With a large number of propositions, the middle ones will be hardest. A similar U-shaped serial position pattern is a common finding in human memory research (Glanzer & Cunitz, 1966). Evidence that the semantic system indeed shows U-shaped performance is given in Section 7.2.

6.3 The comprehension, prediction, and production system

So far we have focused on the semantic system, which accounts for the upper half of the network in Figure 6.1. We turn now to the comprehension, prediction, and production (CPP) portion of the system, which accounts for the lower half and is reproduced in Figure 6.4. The CPP system contains 1.7 million links, giving the complete CSCP model a total of 2.9 million links.

6.3.1 Comprehension and prediction

The main site of input to the CPP system, as one might guess, is the word input layer. Incoming words are represented using a static, distributed, phonological encoding, which was explained in Section 5.4. Each word is presented in its entirety on a single time step, regardless of how many syllables it may contain. Although an earlier version of the model used a single-bit, localist word encoding, a phonological representation was used here because it enables the network to take advantage of any information that can be extracted from phonology. For example, unlike a localist encoding, the phonological encoding assigns morphological variants of a word very similar representations. Word endings can also be a valuable, although not perfectly reliable, clue to noun number or verb form.

Activation flows from the word input layer, through a hidden layer for recoding, and into the comprehension gestalt layer. The comprehension gestalt is the bridge from syntax to semantics. In addition to the bottom-up input, it also receives a delayed self-projection as well as one from the message layer. Thus, the comprehension gestalt is responsible for incorporating the current word into the developing interpretation of the sentence. Its ability to do so will depend on its "knowledge" of both syntax and lexical semantics. The representations formed at the comprehension gestalt will necessarily include aspects of both the syntactic and semantic structure of the current sentence. Incidentally, the term *gestalt* has been borrowed from the sentence gestalt layer of the St. John and McClelland (1988) model, which plays a similar, but not identical, role. The term is used here, as it was presumably used there, to reflect the fact that these areas encode the sentence as a whole, including syntactic, lexical, and semantic aspects.

The message layer is, conceptually, the same message area used in the semantic system and, therefore, is the interface between the two systems. The main goal of sentence comprehension in the CSCP model is to produce a representation at the message layer that is identical to that which would result from encoding the propositions representing the meaning of the sentence. Thus the target of comprehension is the compressed propositional message produced by the encoder half of the semantic system, and is not directly queried during training. The assumption in this mechanism is that the true message is available in advance of comprehension to provide a training signal. The reasonability of this and other assumptions implicit in the model are discussed further in Section 6.6.

Rather than *cross entropy*, the error measure used to assess the mismatch between the correct message and the target message is *sum squared* error, which is defined as follows:

$$\text{sum squared error} = \sum_i (t_i - o_i)^2$$

The reason for this is that the cross entropy measure can grow exponentially large when unit outputs are very close to 0 or 1 but far from the target output. This can result in numerical overflow on a limited precision machine. Even if the error is bounded to prevent overflow, large errors in untrained networks can result in big weight changes that produce even larger errors and the eventual explosion of weights in the network. Although this is not a problem for most networks and somewhat smaller versions of the model had no trouble, the current model consistently encountered this problem early in training. It may have been possible to devise a version of cross entropy that is completely safe from this problem, but a more immediate solution was to simply use sum squared error.

Prediction

While performing comprehension, the CSCP model also attempts to predict the next word in the sentence. The comprehension gestalt and message layers project to the production gestalt, which also receives a self-recurrent projection. The production gestalt then projects through a hidden layer and down to the prediction layer, which generates a distribution indicating the likelihood of each word occurring next. This same system that performs prediction during comprehension will also be responsible for production, which is discussed in Section 6.3.2.

The word representation used at the prediction layer is somewhat different from the one used in the input. The reason is that, for production, we need to be able to generate a likelihood distribution over possible next words. This is a linear combination of the representations of those words whose weightings are proportional to the likelihood of the words occurring. If we were to use distributed phonological codes, as in the word input layer, a linear combination of them would be an indecipherable mess.

Therefore, as discussed in Section 5.4, a more localist encoding is used for prediction and production. The prediction layer is actually composed of two parts, a *stem* and an *ending*. The stem uses a one-bit-on encoding to represent the phonological base of the word, and the ending uses a separate one-bit-on code to indicate if the word has one of six common endings: -d, -ing, -ly, -n, -o, -z, or nothing.

The two parts of the prediction layer are actually *softmax* groups. A softmax group has a normalization constraint that forces its total activation to sum to 1.0. Rather than using the sigmoid function, the activations of the individual units are computed using normalized Luce ratios (Luce, 1986). The output of unit i is given by $o_i = e^{x_i} / \sum_j e^{x_j}$, where x_i is the unit's net input and j ranges over all units in the group.

The target for prediction is simply the next word in the sentence, and is thus readily available in the environment. As in the message layer, the sum squared error measure is used in providing feedback. Earlier versions of the network used *divergence*, which is related to *cross entropy*, but it, too, suffers from numerical instability.

Semantic influence

Having made the assumption that semantics is often available in advance to serve as a target of comprehension, I might suggest without much controversy that semantics can also serve as an input to the comprehension and prediction system. Having a good idea of what is coming next can bias comprehension and can also provide a strong constraint on prediction.

Semantics exerts its influence on the model by *weak-clamping* the message layer units. When a unit is *weak-clamped* to a certain value, its activation will be a weighted average of that value and the output the unit would otherwise have. The *clamping strength* parameter is a number between 0 and 1 which determines the weighting. A clamping strength of 0.25 means the actual output is 25% the clamp value and 75% the original value. So if the original value were 0.2 and the clamp value were 1.0, a clamping strength of 0.25 would result in a final value of 0.4.

On half of the comprehension trials, no semantic information is provided to the network via clamping at any point. On the other half of the trials, the *message* units are weak-clamped to their correct values with a strength of 0.25. That is, as each word comes in and the message is updated, it will constantly be drawn toward the correct representation. Before the sentence begins, all units in the network are normally reset to activations of 0.5. However, when semantic inputs are provided, the message layer is then initially weak-clamped toward the correct representation. The strength of this weak-clamping may differ from the general clamping strength. A clamping strength of 0.25 is used 20% of the time, the strength is 0.5 20% of the time, and another 20% of the time it is 0.75. The remaining 40% of the time the initial clamping strength is 1.0. In this last case, the semantic units are effectively just set to the correct message representation prior to the first word in the sentence. While processing subsequent words, the message units are clamped with a strength of 0.25, provided the initial clamping strength was greater than 0.

Prior to the first word in the sentence, a *start* symbol is fed into the word input. This allows the network to make a prediction of what the first word in the sentence will be. In the case in which the message units start out strongly clamped, the initial word prediction can be very accurate. Assuming it has been trained well, the model should know which word will start the sentence. As long as the message representation does not drift away from the correct meaning, predictions should continue to be very focused and accurate.

One might notice that there is no self-recurrent projection in the message layer. Instead, the message layer projects down to the comprehension gestalt which projects back to the message layer. The reason for this is that the network could learn to rely too much on any self-recurrent connections in the message layer. Because the message layer is often soft-clamped to the correct representation, there is a lot of pressure for links from a unit back to itself to grow very strong. Although this helps the network perform better when semantics are clamped, it can hurt the network when semantic information is not provided in advance. In the latter case, a network with self-recurrent connections would not rely enough on bottom-up information from the comprehension gestalt in formulating the message. Using indirect recurrence forces the network to integrate information provided by clamping with other bottom-up information and, as a result, avoids over-reliance on self-connections.

6.3.2 Production

The primary motivation for incorporating prediction with comprehension is that prediction is the basis by which the model learns to produce language. When the message is not provided in advance, the network must rely purely on the statistics of the grammar to formulate predictions. However, when the message is provided in advance, a well-trained network should be able to predict precisely the correct words. This is, essentially, what is required for production. A major assumption implicit in the model, therefore, is that the formulation and testing of covert predictions during comprehension is a fundamental mechanism by which we humans learn production. Ignoring its physical requirements, production is, in a sense, the ability to predict what a normal speaker would say to convey the intended message.

Production in the CSCP model can be tested in two ways. All production trials begin with the unit activations in the message layer set to the correct values, and the message layer remains hard-clamped to this intended message throughout the sentence. It is also possible to weak-clamp the intended message, allowing it to drift as the model makes mistakes. However, all of the production results reported here make use of hard-clamping. *Word-by-word production* simply involves predicting the correct words while performing comprehension in the presence of an intended message. On each step, the network's "production" is taken to be the word composed from the most highly predicted stem and the most highly predicted ending at the prediction layer. However, no matter what the network would have produced, the actual next word in the sentence is always fed in on the next time step.

The alternative to word-by-word production is *free production*. The network starts as before, with a fully clamped message and a *start* symbol at the word input. But now, whatever the network predicts to be the first word of the sentence is fed back into the word input on the next time step. This is reflected in the dashed arrow in Figure 6.4. Of course, in transferring the word from the model's output to its input it must be translated from the localist phonological encoding to the distributed phonological encoding. The network then runs for a second time step and predicts, or produces, the second word in the sentence. This continues until the network produces the end of a sentence (a period) or until some maximum number of words is reached.

While the network is engaged in free production, it is monitoring itself and attempting to comprehend what it is saying. Thus, the production system is heavily reliant on the comprehension system. Although the message layer is weak-clamped while producing a sentence, it is not hard clamped so that there is a possibility that the model can stray off message.

Training comprehension and production

In the current version of the model, there is no actual training on free production. All training occurs during comprehension and prediction, with and without the presence of semantic input. During training, there are two sources of feedback for the network. The target for the message layer is the correct message for the sentence, and the target for the prediction layer is the next word in the sentence. Prior to training on the sentence, then, the semantic system must be used to encode the actual message of the sentence.

Like the message encoder, the CPP system is trained using backpropagation-through-time. There is a single forward phase for each sentence, followed by a single backward phase. In the forward phase, each word is presented to the model, beginning with the *start* symbol, and the units update their activations from the word input up to the message and down to the prediction. Error is assessed at the message and prediction layers. The last word in a Penglish sentence is always a period. In reading, this could be a literal period. In listening, this could be thought of as a pause after the sentence. Following the period, the message layer is updated to obtain a final message, but no additional prediction is made.

The backpropagation phase starts with the message layer on the last time step. Error derivatives are backpropagated to the word input layer. The activations in the network are then reset to their state on the next-to-last time step, and the previously recorded output error derivatives are injected at the prediction and message layers. The backpropagation then runs through all of the layers, from prediction to word input. This repeats, running backwards in time, until the first tick has been reached. Note that there is no information backpropagated across the dotted arrow from prediction to word input during this process.

6.4 Training

This section describes the methods used to train the CSCP model. The networks are implemented using the LENS neural network simulator, which I wrote partially in preparation for conducting a large simulation such as this. Some of the important design features of LENS are explained in Appendix A. Because of the shortcuts necessary to reduce the training time for the semantic system, and because of the need to analyze the network in various ways, such as extracting reading times, a lot of customized code was required to extend LENS in order to implement the model.

The two components of the model, the semantic system and the comprehension, prediction, and production (CPP) system, were trained separately. All weights in the network were initialized with a value drawn at random from a flat distribution between -0.3 and 0.3. The only exception is the incoming links to the query gestalt layer, which were initialized to a range of -0.2 to 0.2. Although the conventional wisdom has been that neural networks should be initialized with very small weights (± 0.01 or less), I have found that simple-recurrent networks operate best when initialized with fairly large weights (Rohde & Plaut, 1999), and there are some theoretical justifications to support this. If a group has many incoming connections, smaller weights should be used, but only slightly smaller.

Both the semantic and CPP systems were trained using a batch size of 5 sentences. That is, link weight error

Epoch	Rate	Epoch	Rate
1	0.200	9	0.020
2	0.100	10	0.020
3	0.060	11	0.020
4	0.040	12	0.015
5	0.030	13	0.015
6	0.030	14	0.010
7	0.025	15	0.010
8	0.025	16	0.001

Table 6.1: Learning rates used in training both the semantic and CPP systems.

derivatives were accumulated over 5 sentences before the weights were updated. The batch size is an important tradeoff in training a network such as this. Because the error surface is so complex, using too few sentences per batch would result in constant changes in direction and a lot of time wasted doing unnecessary updates. A batch size that is too large will also hinder learning because the weights are not updated frequently enough. In general, with sentence processing tasks, batches of 5 to 20 are often reasonable, with smaller batches used for longer, more complex sentences.

Weight updates were performed using *bounded momentum descent* with a momentum of 0.9. Bounded momentum descent is a technique I developed along with LENS. Ignoring momentum for a minute, the idea behind gradient descent is that one should take a step in weight space in the *direction* of steepest descent. But the traditional implementation of steepest descent takes a step that is not only in the direction of steepest descent but whose horizontal distance is proportional to the magnitude of the slope in that direction. This does not quite make sense. Consider this analogy. If you were walking down a mountain and the slope were gradual, you would be taking dainty little steps, and not getting down very quickly. But if the slope were steep, you would be taking huge strides and perilously plunging down the mountain. When using traditional steepest descent, especially with momentum, one normally needs a small learning rate with no momentum initially when the mountain is steep, and then a larger rate and momentum once the slope has flattened out, followed by gradually smaller learning rates as the minimum is neared. With this method, some error measures, like sum squared, can tolerate fairly large learning rates, while cross entropy and divergence require much smaller ones. The best learning rate depends on the size of the network and the difficulty of the task.

Bounded gradient descent, on the other hand, takes a step whose horizontal distance is no larger than the learning rate, no matter how steep the slope. So whether the slope is gradual or steep, you will always use the same stride, horizontally speaking, although more vertical progress will still be made when the slope is steep. The learning rate determines the maximum size of the step and, following normalization, the momentum term is then added to the weight step as in standard momentum descent. With bounded momentum descent, more consistent learning rates can be used with different error measures, networks, and tasks. There is also less need to adjust the learning rate early in training. It is possible to leave momentum constant and start with a high learning rate. The learning rate just needs to be gradually reduced over the course of training for best results.

Sixteen training sets were generated using the Penglish grammar, each set containing 250,000 sentences, for a total of 4 million training sentences. The semantic system was first trained for 50,000 weight updates on each of the training sets. With a batch size of 5, that is a total of 250,000 sentences on each set. Each round of 50,000 updates on a single training set is considered an *epoch*. Therefore, there were 16 total epochs. In order to save some time, any sentences with more than 8 propositions were filtered out during training. If the end of a training set was reached before 50,000 updates had completed, training resumed at the start of the set. Therefore, some sentences (a bit under 4%) were seen more than once by the network.

The learning rate started at 0.2 for the first epoch and then was gradually reduced over the course of learning. Table 6.1 shows the learning rate used for each epoch of training. High learning rates are useful initially because they enable the network to perform broad reorganizations of its representations. Later in training, a smaller learning rate enables the network to focus on the more subtle statistical information in the environment without disturbing the broad organization. This is especially true of the last epoch, for which the learning rate is much reduced.

Once the semantic system was done training, the CPP system was trained, with the weights in the semantic system remaining fixed. Once again, the model was trained for 16 epochs, each containing 50,000 updates, or 250,000 sentences. The same progression of learning rates was used as before. However, in this case, training began with sentences of limited complexity, and the complexity increased gradually. On the first epoch, only sentences with one or two propositions were allowed. On the next epoch the limit was increased to three. On each successive epoch, the limit increased by one more until, on the seventh and all later epochs, all sentences with 8 or fewer propositions were allowed. Thus, the environment fairly quickly reached its final level of complexity.

Those who are familiar with my earlier work on sentence processing (Rohde & Plaut, 1997, 1999, in press), might be surprised at the use of staged input here. In those earlier papers, we argued that staged input can be a significant hindrance to learning in a prediction network. However, we were careful to point out that the same conclusion may not apply to a comprehension network:

> If children do, in fact, experience simplified syntax, it might seem as if our findings suggest that such simplifications actually impede children's language acquisition. We do not, however, believe this to be the case. We have only been considering the acquisition of syntactic structure (with some semantic constraints), which is just a small part of the overall language learning process. Among other things, the child must also learn the meanings of words, phrases, and longer utterances in the language. This process is certainly facilitated by exposing the child to simple utterances with simple, well-defined meanings. (Rohde & Plaut, 1999, p. 98)

In this case, not a lot of research went into the decision to use staged inputs. One preliminary network was trained with progressive inputs and one was trained with full and the former learned a bit better. Therefore, staged inputs were used in training the final versions of the model.

Training a single epoch of either the semantic or CPP system of a network requires about two days on a 500MHz Alpha 21264 processor. The total training time for the model was, therefore, about two months. Because of the length of training, it was not possible to carefully optimize the performance of the network, and no attempt was made to tweak its behavior to achieve a certain result. With the advent of faster machines, training time should be reduced to a few weeks and more experimentation will be possible on the architecture, training methods, and environment of the model.

Three separate networks were trained, starting with different random initial states. Each network had its own semantic system and its own CPP system. The reason for training three networks was to test the extent of individual differences in the model. For convenience, the three networks will be referred to as **A**dam, **B**ert, and **C**had.

6.5 Testing

A separate testing set of 50,000 sentences was generated independent of the training sets. When needed, subsets of this testing set were often used for running various experiments. Not all sentences in the testing set are novel to the networks. On average, 33.8% of the testing sentences also appeared in one of the training sets. However, nearly all of the familiar sentences had just one or two propositions. Figure 6.5 shows the percentage of familiar sentences in the testing set as a function of the number of propositions in the sentence. Of sentences with four propositions, only 2% were familiar. 0.5% of those with five propositions were familiar, and no sentences with six or more were familiar.

6.5.1 Measuring comprehension

When testing the model on comprehension, the message is not provided by weak-clamping the message layer. That would, of course, be cheating. All of the words in the sentence are fed in until a final message representation has been formed. Then the decoder system is used to query this representation to see if the model can answer all possible questions about the propositions composing the true sentence meaning.

Recall that each query requires the network to fill in a missing part of a proposition. Two measures can be used to assess whether the network has correctly filled in the answer: a *strict* criterion and a *multiple-choice* criterion. In

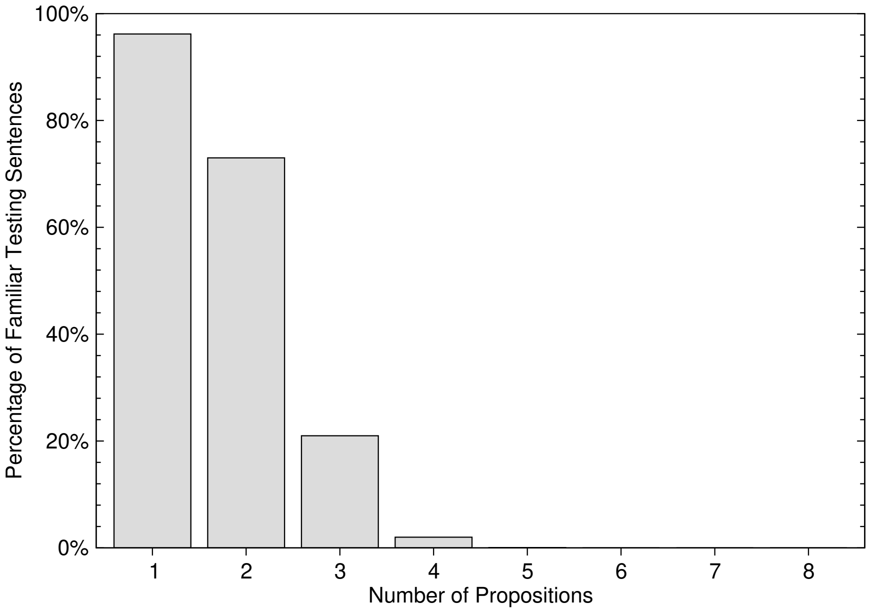


Figure 6.5: The percentage of familiar (trained) sentences in the testing set.

the strict criterion, all bits in the answer must be on the correct side of 0.5. That is, if the bit should be a 0, it must be less than 0.5 and if the bit should be a 1, it must be greater than 0.5. Chance performance on the strict criterion is essentially zero. In order to guess the right answer, the network would have to guess the correct values of either 25 or 134 bits. In the former case the chance is one in 33 million and in the latter it is, well, let's just say it could never happen.

Measuring performance using the strict criterion is informative, but it is not necessarily a good model of how the comprehension performance of human subjects is measured. Most experiments seem to test comprehension using true/false questions. The difficulty of such questions and the type of information probed can vary widely. Given the sentence, "*When the owner let go, the dog ran after the mailman*," an easy true/false probe might be, "*The dog ran after the doctor?*" A harder question might be, "*The dog was attached to a leash?*" In its present design, neither type of question could directly be asked of the model, although the latter certainly goes beyond the level of inference expected of it.

To answer the former type of question, the subject presumably makes some comparison between the proposition in the question and the information derived from the sentence. If there is a serious mismatch, the answer will be false. One way to come closer to this sort of comprehension test is to ask the model a multiple-choice question. That is, a probe is given and the network's response must be closer to the correct answer, according to cross-entropy error, than it is to any of a set of other possible answers, or distractors. In testing the CSCP model using the multiple-choice criterion, the number of distractors is variable, but four is the number used in the following analyses. The distractors include any other constituents of the meaning of the sentence that differ from the correct answer but are of the same semantic class. For example, if the correct answer is an object, then any other objects in the sentence may be used as distractors. If there are not enough such distractors, then the remainder of the distractors will be drawn at random from all possible items of the same semantic class. The semantic classes include objects, actions, properties, and relationships. Using only distractors of the same semantic class is harder for the model, because it would almost never choose a distractor of a different class from the correct answer.

In the case of the sentence above, the question might be, [ran after, theme, ?]. The correct answer is *mailman*, and the within-sentence distractors would be *owner*, and *dog*. To these, two other distractors will be added, perhaps *girls* and *binoculars*. The question, then, is whether the network's response is closer to *mailman* than it is to any of the distractors. The function used to measure the distance between vector representations in this case is sum squared error. When using four distractors, the chance performance is 20% correct.

6.5.2 Measuring reading time in an SRN

Much of the interesting psycholinguistic data on sentence processing involves measurements of reading time, using either self-pacing or eye-tracking. In order to model such data, we will need a word-by-word measure of the model's reading time. Because the model spends just a single step of simulated time on each word, we cannot get reading times directly from the settling time of the model. Several other researchers have encountered this same problem with SRNs and have devised some different solutions.

Christiansen and Chater (1999a) (see also MacDonald & Christiansen, in press) developed a reading time measure for SRN prediction networks called the *grammatical prediction error* (GPE). Essentially, it bases reading time on the accuracy of the network's next-word predictions. Activation of the output units is segregated into *hits*, *misses*, and *false alarms*. At any point in a sentence, some words, or word classes, will be grammatical continuations and some will not. *Hits* is the sum of all activation over output units representing grammatical next-words. *False alarms* is the sum of all activation over ungrammatical words. *Misses* is a measure of the degree to which the activations of grammatical words are less than they should be given the overall lexical frequency of the word. The GPE reading time is computed as follows:

$$\text{GPE} = 1 - \frac{\text{hits}}{\text{hits} + \text{misses} + \text{false alarms}}$$

This measure is problematic for use in the current model for several reasons. Principally, it is only a model of prediction error—it does not take into account effects in comprehension, which may also have a major impact on reading time. Furthermore, the measure is based on the assumption that the grammar of the language is entirely syntactic, with no lexically-specific semantic effects. For example, when the next word must be a noun, the GPE computation assumes that the likelihood of each noun will be in proportion to the overall frequency of the noun across the language. Therefore, it does not allow for context effects, which are pervasive in both English and Penglish, and it seems likely that the GPE measure would be insensitive to many of the lexical frequency and semantic constraints that have been shown to affect human processing of ambiguous sentences.

Another problem with this measure is that it presupposes knowledge of the grammar that the reader may not actually possess. Reading delays in humans are a result of processes internal to the reader. Logically, any information that affects those delays must be information that the reader has knowledge of or access to. In the case of a network model, it must be information that the network has learned. However, computing the GPE involves classifying certain predictions as either grammatical or ungrammatical and that is a judgment based on *external* information about the grammar. The network itself may have no understanding of what is and is not grammatical and thus will not necessarily be sensitive to its own mistakes. Consider, for example, the case of a network with completely randomized weights that pays no attention to its inputs and gives the same prediction for every word in all contexts. One should expect that a naive reader such as this would have reading times that are uncorrelated with the grammar of the language. However, the GPE, because it relies on external knowledge of the grammar, will predict different reading times for the network based on where it is in the sentence. If many words are possible at a certain point in the sentence, with roughly equal likelihood, the GPE will be low. If only one or two words are possible, the GPE will be high. Thus, the GPE will ascribe sensitivity to the language that this randomized network could not possibly have.

A quite different method for extracting reading times from prediction SRNs has been developed by Tabor, Juliano, and Tanenhaus (1996), Tabor et al. (1997) and Tabor and Tanenhaus (1999). Their method, known as the *visitation set gravitation* (VSG) model, derives reading times by simulating a process of settling in a high-dimensional gravitational space. The activation states of the hidden layer that occur at some point in processing a large number of sentences are all plotted as points (which I will call *stars*) in a high-dimensional space and each is given a gravitational field. To determine the reading time for a word in a test sentence, the hidden unit state that occurs while processing the word is plotted in the high-dimensional state. This is considered to be a point mass that moves under the effect of gravity from all of the stars. The reading time is proportional to the number of steps the test mass takes before it sharply changes direction. The idea is that points that start out near a dense cluster should settle faster than points that start out in a less populated part of the space. This simulation is meant to reflect what might happen in a fully recurrent dynamical network that must actually settle to a stable state.

There are several reasons this may not be an appropriate method for use with the CSCP model. The fact that it was developed for a model of word prediction, rather than semantics, is again problematic, but is not too serious.

In principle, one could extend the model by composing the high-dimensional vectors from the activations of several different layers in the network, some involved in prediction and some in comprehension. A more serious problem is that it is just not clear how well this method really works in predicting word-by-word reading times because its behavior is so dependent on the density of the stars and the parameters that govern them. One would like to see a broad analysis of its behavior across many sentence types, both unambiguous and ambiguous.

It seems possible that, even if a word is at a difficult point in a sentence, the test mass may start very close to a star and therefore experience a rapid change in direction where we would expect a long reading time. This would be a particular problem if the sentence being tested (or at least the initial substring of the sentence) happened to have also been used in positioning the stars in the first place. Unless noise is introduced, the test mass would fall directly on a star, which would likely result in a very short settling time. The VSG model may not be sensitive only to the overall density pattern of the stars but also to interactions with specific, local stars and thus may have strange or unpredictable behavior at times.

The VSG model has so far been used only with languages having very simple grammars, with perhaps 10 different parsing states. In this case, the stars tend to fall into a few dense clusters. This pattern seems to be important to the results that have been reported using the method. The Penglish language, however, presumably involves many thousands of different states and the clustering of those states is likely to be far less well-defined. This creates two problems. One is that the resulting settling times may be even more sensitive to local structure, or the random presence of nearby stars, than to the overall pattern of star density. The other problem is the matter of computational efficiency. A more complex language will require many more sample sentences to build an accurate estimate of the density structure of the language. This means many, many stars. The CSCP model also has larger hidden layers. Rather than 10 dimensions, we'll need 500 or 1,000 dimensions. Having many stars and many dimensions will result in an extremely slow settling process.

6.5.3 A theory of comprehension difficulty

Because of the limitations of these earlier methods, particularly in their applicability to a comprehension network, a new technique for measuring reading times was developed for the CSCP model. This technique is based on a novel theory of comprehension difficulty, and it seems appropriate to begin by summarizing the theory. The idea is essentially that human reading times reflect a combination of many specific factors operating at different levels of abstraction. The following is a list of some of the major factors I hypothesize to have an influence on reading times. Other accounts of reading time or comprehension difficulty have sometimes invoked one or two of these, but have rarely involved more than that. In identifying these factors, I have relied on an assumption that the sentence processor is a dynamical system and tried to use my intuition about the behavior of such systems, given the requirements of sentence comprehension, to infer some aspects of words and sentences that may affect the difficulty of comprehension, and hence reading time. As this theory is still in development, all rights to future modification are reserved.

- **Word length.** There is good evidence that longer words take longer to read, although the effect is not necessarily a linear function and may be influenced by other factors such as the regularity of spelling or pronunciation of the word and the presence of common inflectional suffixes, like "*-ment*," which may be read more quickly. Note that the effect of word length may be isolated to the processes of word recognition and may not actually impact comprehension difficulty.
- **Overall lexical frequency.** It is well known that more frequent words, at least in isolation, are read more quickly, all other things being equal. Although this is specific to reading, there may be a parallel effect in recognition of spoken words. Presumably, overall lexical frequency has its main effect at an early stage in processing.
- **Contextual lexical frequency.** Although it has not been well established, the likelihood that a word appears in a particular context may have a direct effect on its reading time that can be isolated from semantic effects. For example, *sugar* may be read more quickly in the sentence, "I like my coffee with cream and *sugar*," than in the sentence, "To make the cake you'll need lots of *sugar*." In both cases it is semantically reasonable, but it can be more easily predicted in the former.
- **Semantic change.** The bigger the effect a word has on the overall meaning of the sentence, the longer one will spend in reading it. Principally, nouns and verbs are major contributors to sentence meaning. Reading one, particularly a verb, will result in a relatively large change in the semantic representation and thus a longer settling time.

As far as I know, this effect has not been validated experimentally. However, the failure to find such an effect may be partially due to a problem in the way reading times are processed.

In most reading time experiments, raw reading times are converted to *residual* reading times by attempting to factor out the effect of word length. This is generally done by computing the average reading time as a function of word length for each subject and fitting this with a linear regression. The average reading time predicted by this regression function is then subtracted from the raw reading time of a word to produce a residual time. However, this process allows a potential confound between word class and word length. Longer words also tend to be closed-class, including nouns, verbs, and adjectives. Thus, the conversion to residual times may be removing an effect of word class or lexical semantics in addition to controlling for word length. It would be interesting to investigate this possibility by computing separate regressions for different word classes for a reading-time experiment. Then again, there may still remain confounds of frequency and other factors.

- **Semantic reasonability.** Semantic anomalies, "I like my coffee with cream and *salt*," may have an effect on reading time that is separable from the effect of the low predictability of the anomalous words. This can occur at a lexical level or possibly as a result of higher-level semantic confusion. This might be considered a special case of semantic change, but it seems reasonable to draw a distinction.
- **Ease of syntactic and semantic integration.** During sentence comprehension, new elements of the sentence must usually be integrated with previous material to derive a complete meaning. This integration will be harder if the previous material is less highly activated or plays a diminished role in the current internal state of the model. This is particular true in cases where integrations take place over a span of intervening material. In general, the longer the distance between the elements involved in the integration, the less active the first element will be and the longer the integration will take. Gibson refers to this as the effect of *locality* and has made it a principal component of his *dependency locality theory* of comprehension difficulty (Gibson, 2000).
- **Projecting upcoming structure.** At certain points in a sentence, the comprehender can predict that some complex structure is on its way that may drastically change the overall representation of sentence structure and/or meaning. Reorganizing the current representations of the sentence in preparation for this new structure could cause delays. For example, reading "The boy that…" may lead one to expect a relative clause and to slow down a bit in preparation for it. Similarly there may be a longer pause on the verb in "When the king slept…" than after, "When the king killed…" because, in the latter case, a direct object might be expected while in the former case an entirely new clause is expected. This factor is similar to the *storage cost*, which is another major component of Gibson's theory (Gibson, 2000).
- **Representing multiple contingencies.** In cases of ambiguity, it may be necessary for the system to represent multiple possibilities in parallel. If one possibility is dominant, this may have little effect on processing time, because fewer resources are allocated to the unlikely possibilities. But if the perceived likelihood of the possibilities is roughly equal, there will be an ongoing competition that may result in slowdowns. Such a competition is a major component of the Spivey and Tanenhaus (1998) model of ambiguity resolution.
- **Resolving ambiguities.** Most models of comprehension difficulty allow for some sort of slowdown at or just following the point of disambiguation after an ambiguous region. Although the terminology may not be standard (Lewis, 1998), one can distinguish between three degrees of ambiguity resolution, *cleanup*, *repair* and *reanalysis*. Cleanup is relatively benign and may incur little processing difficulty. It occurs when the correct interpretation is also the one that was preferred and most highly activated by the system. At that point, resolving the ambiguity is simply a matter of purging the other parses from the representation. When the correct parse was still in the running but was not the preferred one, resolving the ambiguity is more difficult, and involves promoting the correct interpretation as well as purging the preferred one and any others. There is, of course, a smooth gradation between cleanup and repair.

  Reanalysis, on the other hand, involves actually going back and re-processing the sentence at the word level. This might be done by actually re-reading part of the sentence, if possible, or by replaying the sentence from an internal buffer or phonological loop. It is hoped that the CSCP model is capable of cleanup and repair, but it has been given no mechanism for reanalysis.
- **Capacity limitations.** Capacity limitations of some sort are widely thought to play a role in sentence processing (King & Just, 1991; Just & Carpenter, 1992). If one thinks of the sentence processor as filling a limited size memory bank with discrete facts, capacity limitations will be a function of the amount of available memory. Alternatively, it could be a function of how small those facts are, and, thus, how efficiently the system can use its available memory.

When the memory is filled, slowdowns are expected. This sort of model would seem to predict that memory plays little role until the limit is reached at which point a complete breakdown would occur, or, at least, a discrete loss of some of the information.

Thinking along connectionist lines, I view the use of memory by the system a bit differently. Memory should be viewed not as a bank of slots but as a high-dimensional representational space. Information stored in memory is not composed of discrete elements but of a single, complex distributed representation, with portions of the information compressed to varying degrees. The more information that is to be stored, the more it must be compressed. The more compressed the information, the greater the chance of its being lost and the harder it will be to access the information in a useful form. As the amount of information stored stresses the available capacity, errors and slowdowns will occur as a result of the need for greater compression. However, those problems will tend to come on gradually and may even be detectable long before the overall capacity is reached.

- **Wrap-up effects.** Finally, it is a common finding in reading time experiments that subjects dwell for a long time on the last word in a sentence (Just & Carpenter, 1980). There may be several explanations for this. One is that the realization that the sentence is over acts as an additional input that causes further restructuring of the internal representation of sentence meaning. Although semantic information must be retained, much of the syntactic information involved in parsing can be discarded. The fact that the sentence has ended may also resolve some open ambiguities. Furthermore, the subjects may also be engaged in explicit reanalysis of the sentence, replaying it in their minds. This may be especially true in cases when they know a difficult comprehension question is one key-press away. It may be that the wrap-up effect should not really be considered a factor in and of itself, but that processes governed by several other factors all happen to kick in at the end of a sentence.

The current theory proposes that each of these factors has an individual effect on comprehension difficulty, which indirectly affects reading times. Word length and lexical frequency, however, may have a direct influence on reading time. Some of these factors may be highly correlated so that it may not be possible to isolate their effects experimentally. Simply separating the effects of word length, word frequency, and a word's semantic properties seems quite difficult, though not impossible. For the time-being, I will remain agnostic about whether specific factors ought to have independent effects or will interact in a non-linear way. But I will venture to guess that some degree of interaction will occur, and, in particular, capacity limitations are likely to interact with the representation of multiple contingencies and ambiguity resolution.

The issue of interactions is germane to the question of how comprehension difficulty relates to reading time. There seems to be a general assumption in the field that reading times, corrected for word length and perhaps word frequency, directly reflect local comprehension difficulty. However, the relationship between difficulty, if that can indeed be quantified, and reading time may be rather non-linear. This non-linearity has the potential to mask real interactions or create apparent interactions between factors. While, mechanistically, two factors may have completely independent effects on comprehension difficulty, their combined influence on reading time may be disproportionately large or small due to a non-linear transformation.

There are other reasons to think that reading times may not directly reflect comprehension difficulty. The effects seen in reading time are notoriously delayed, especially in self-paced reading involving key-presses. Even where a point of difficulty is clearly defined, the slowdown in reading may not reach full strength until the next word and may be spread over several succeeding words. This could reflect the fact that comprehension difficulty is indeed delayed, or it could reflect a delay between difficulty and its expression in reading time, possibly due to a slow signal to change a habitual key-press rate.

Reading times are also subject to strategies. Although there may not be much direct evidence of this, this seems clear from observing subjects in self-paced reading experiments and will likely be borne out in controlled studies. For example, if the sentences used in an experiment tend to be easy and are followed by easy comprehension questions, subjects will go faster overall and may experience large garden-path effects when there is, in fact, a difficult sentence. If most sentences start in a similar fashion, subjects will tend to read the beginning of sentences more quickly and then slow down at the first sign of difficulty. If the length of a sentence is visible from the start, as in eye-tracking or masked self-paced reading, subjects may adopt a particular pace early on that is dependent on the length of the sentence. When sentences are consistently hard but are varied, subjects will proceed deliberately and will be more sensitive to possible ambiguity and show reduced effects of those ambiguities. When sentences are too hard or are potentially ungrammatical, subjects may simply give up and read quickly. Thus, reading time may not reflect comprehension difficulty, especially across studies with different experimental and filler items.

It is worth stressing again that the above list of factors potentially affecting comprehension difficulty is not meant to be complete. In particular, several factors have been left off that are unrelated to the current operation of the CSCP model because of its limitations and those of the Penglish language. A factor worth mentioning is the difficulty presented by anaphora, including pronoun resolution, or referential processing more generally. Another one is the problem of establishing clausal coherence relationships that are necessary to make sense of any complex discourse. I do not, however, consider extra-sentential context to be a factor in and of itself, although context can influence other more proximal factors, such as semantic reasonability and the resolution of ambiguities.

6.5.4 The CSCP reading time measure

We are finally in a position to explain how reading times are actually measured in the CSCP model. The reading time measure used in this study will be referred to as *simulated reading time* (SRT). The SRT is a weighted average of four components, two based on prediction error and two based on semantic representations.

The first two components measure the degree to which the current word was expected. Recall that the prediction layer actually contains two parts, the *stem* and the *ending*. The stem has one unit for each base form of a word and the ending has one unit for each of several common suffixes. The first two components of the SRT reflect the prediction error of these two groups on the previous time step. The measure used is actually the divergence error, which is equivalent to the negative log of the activation of the correct output unit. The less active the correct unit, the higher the error will be. The divergence is calculated separately on the two prediction groups.

These two components of the SRT presumably reflect overall and contextual lexical frequency. They may also reflect, to some extent, semantic reasonability and the ease of syntactic integration. Points in a sentence where long integrations occur, such as the verb following a subject-modifying embedded clause, may also be points where predictions are difficult and prediction error for the model will be higher than would be expected from the entropy of the true distribution.

The next component of the SRT is the change in the message that occurred when the current word was read. This is actually the root mean squared difference between the previous activation and the current activation of each unit in the message layer. This is equivalent to the Euclidean distance between the old representation and the new one in vector space. The bigger the change in the message, the larger this value will be. Message change probably reflects several different factors, including semantic change, ease of semantic integration, projecting upcoming structure, representing multiple contingencies, and resolving ambiguities. It may also be influenced by capacity limitations. As the message layer nears its representational limit, large changes in encoding may be necessary. This measure is really a surrogate for the time that a truly interactive system would take to settle to a stable message representation.

That a representational change in the message layer directly reflects message change is actually somewhat surprising. In the CSCP model, messages are generally quite sparse. They tend to begin with most units with activations near zero. As the sentence comes in and the message grows, the message representation becomes gradually more dense, although not always monotonically so. This seems perfectly reasonable, but is not the only way that a connectionist model could operate. It is entirely possible that a somewhat different model might find a solution in which the message representation jumps around radically in high-dimensional space with each word. Such is the case with earlier prediction models which sometimes encoded syntactic embeddings by cycling around in representational space (Elman, 1991).

The final component of the SRT is the average level of activation in the message layer. As mentioned, the message representation usually starts out quite sparse and then becomes gradually more dense. One might hypothesize that as the message representation grows in complexity, it will decrease in flexibility and changes to it must be made more slowly to avoid loss of old information. Thus, the activation in the message layer is related to capacity limitations and may also reflect the maintenance of multiple contingencies.

To produce the actual SRT measure, the four components are multiplied by scaling factors to achieve average values of close to 1.0 for each of them and a weighted average is then taken. Table 6.2 shows the scaling factors for the four components and the weighting used in computing the SRT. The message change accounts for 50% of the SRT, while the message activation and the predictions together each account for 25%. Because it is a linear combination of normalized values, the SRT has an average value that is very close to 1.0 as well. It ranges from around 0.4 for easy words to 2.5 or more for very hard words.

Component	Scaling Factor	Weighting in SRT
Word Stem Prediction	0.365	22%
Word Ending Prediction	2.646	3%
Message Change	0.666	50%
Message Activation	9.434	25%

Table 6.2: The components of the simulated reading time (SRT).

Although the SRT is called a measure of reading time for convenience, it should be understood that it is really a measure of comprehension difficulty. The SRT does not reflect certain factors that are believed to play a role in reading times but not in comprehension difficulty. To begin with, the SRT does not show the delayed effects that characterize self-paced reading times. Thus, while a self-paced reading experiment might find a delayed slowdown on the words following a point of disambiguation, the SRT would show an immediate effect. Eye-tracking methods generally differ from word-by-word self-paced reading in that eye-tracking permits parafoveal preview of the next word. This will probably have the opposite effect of redistributing some of the reading delays to the previous word. Because the model is not given any preview of upcoming material, it will not show such effects either.

Furthermore, the SRT, like the CSCP model in general, is not subject to strategies. The model will not read faster because the sentence appears short or because it is approaching the end. It will also not begin to read faster if it sees a lot of easy sentences. At least, certainly not if tested with the learning mechanism disabled. If the learning mechanism is engaged and the model is exposed to a number of similar sentences, it will no doubt adapt to that sentence type, resulting in better predictions and faster reading times. This could be viewed as syntactic priming expressed in comprehension.

The SRT also cannot show effects of word length because the model's word input uses a phonological encoding with no temporal structure. Thus, it is reasonable to think of the SRT as a residual reading time. In doing so, however, it is important to remember that the standard method of computing residual reading times may confound word length with word class and frequency and thus dilute the effects of those other factors as it tries to eliminate word-length effects. The SRT also does not reflect offline processes such as those involved in reanalysis and wrap-up effects. Even when doing self-paced reading with masking of previously viewed words, subjects may pause at a point of difficulty and replay part of the sentence in their heads from a phonological buffer. A more sophisticated model might be capable of such *post-analysis*, but the CSCP model is limited to proceeding sequentially through a sentence. Therefore, the SRT is expected to show much shorter delays at "garden-path" points that trigger reanalysis in subjects. This is particularly true at the end of a sentence.

6.5.5 The CSCP grammaticality rating measure

Aside from reading times, another popular method for assessing sentence difficulty is asking subjects to judge whether sentences are grammatical or to rate their grammatical complexity on a scale. One problem with such measures is that it is not clear what properties of the sentence subjects are actually basing their responses on. Presumably subjects are not strictly applying the rules of grammaticality taught in school, such as the one that should have told me not to end the previous sentence with a preposition. In fact, subjects are sometimes instructed to disregard such rules. It is often supposed that subjects will attempt to apply the syntactic rules of everyday language in formulating their judgments. But, in practice, there also seems to be an important semantic element to grammaticality judgments. "Colorless green ideas sleep furiously," although syntactically valid, would probably rate fairly low on a test of grammaticality.

The CSCP is not able to explicitly provide its own grammaticality ratings. Therefore, as with reading times, a method was developed to extract such ratings from other aspects of its behavior. Because I am presuming that human grammaticality judgments are based on both semantic and syntactic factors, the model's judgments are composed of two components: prediction accuracy and comprehension performance.

The accuracy of the model's word predictions can be an indicator of the syntactic complexity of a sentence. Among grammatical sentences, more syntactically complex ones, such as those with center embeddings, will tend to contain points at which the prediction is difficult. And if there is a clear syntactic violation in a sentence, due to missing,

Word	Stem Activation	Ending Activation	Prediction Error
The	0.45245	0.82659	0.984
lawyer	0.06239	0.88429	2.897
that	0.01121	0.61550	4.976
the	0.06554	0.73166	3.038
cop	0.04471	0.90954	3.202
asked	0.03009	0.11395	**5.676**
believed	0.00090	0.00048	**14.655**
the	0.22504	0.94134	1.552
father	0.07398	0.87678	2.735
.	0.02147	0.64679	4.277

Table 6.3: An example of how the prediction error is calculated for the purpose of computing grammaticality ratings.

additional, or transposed words, the model's predictions at that point are likely to be very far off the mark. So an extremely bad prediction may signal a syntactic violation. However, if there is a true violation, it is also the case that the model will probably not recover immediately. It will become confused as to where it is in the sentence. This is likely to cause prediction errors on the next word or the next few words, until the model gets back on track.

Therefore, the prediction component of the grammaticality rating for a sentence involves the point in the sentence at which the two worst consecutive predictions occur. Numerically, the prediction error on a particular word is computed as the negative log of the activation of the prediction layer output unit representing the word. If the activation of the unit is very low, its negative log will be high. Of course, there are actually two parts to the prediction layer, the stem and the ending. Each word is represented by one unit in the stem and one in the ending. So in reality, the prediction error is computed using the product of the two unit activations. The prediction error for a pair of consecutive words is just the sum of the errors on the two words. The prediction error for a sentence is the maximum error on that sentence for any pair of consecutive predictions.

To help clarify this a bit, Table 6.3 shows the prediction error for the sentence, "The lawyer that the cop asked believed the father." The first column of numbers is the activation of the word stem prediction unit corresponding to the word at the left. High values indicate successful predictions and low values reflect surprise. The second column of numbers are the activations of the word ending prediction units. The last column is the prediction error on that word, which is the negative log of the product of the two activations. The numbers in bold are the consecutive predictions with the highest sum, which happens to be 20.33. Although they are grammatical, the model really does not like center embedded object relatives. It is not always the case that the two consecutive words with highest prediction error are also the two words with overall highest error. But for a sentence with a true syntactic violation, that is often the case.

The other part of the grammaticality rating is simply the average strict-criterion comprehension error rate on the sentence. This is intended to reflect the degree to which the sentence makes sense. It is true that using a measure of this form is not a plausible model of what human subjects are doing. For one, human subjects do not have available to them a list of comprehension questions, let alone their answers. Presumably, we assess comprehensibility with a slightly difference process. For example, we may ask ourselves a series of questions: "Does each constituent in the sentence have a clear role? Is the constituent, semantically and pragmatically, able to fill that role? Are there any conflicts between those roles? Are there roles left unfilled that require an explicit constituent?" The current model does not have the ability to pose such questions to itself. However, it is likely that the ability to answer such questions and the ability to answer fill-in-the-blank comprehension questions will draw on the same knowledge. Therefore, for simplicity, the standard comprehension questions are used as a proxy for a more plausible form of comprehensibility self-assessment.

The following formula is used to combine the two components just discussed—the sentence prediction error and the sentence comprehension error—into a single measure of ungrammaticality, known as the *simulated ungrammaticality rating* (SUR):

$$\text{SUR} = (\text{PE} - 8) \times (\text{CE} + 0.5)$$

where PE stands for the prediction error (summed over the two worst consecutive predictions) and CE is the percentage of comprehension questions answered incorrectly by the strict criterion. The SUR is the product of two components. The first is proportional to the PE. Eight is subtracted from the PE because, even for very simple sentences, the PE is usually at least 9. This makes the SUR more sensitive to PEs in the critical 10-15 range. The second component is the CE + 0.5. The CE ranges from 0 (no errors) to 1 (incomprehensible). The amount 0.5 is added so the CE will have the effect of scaling the SUR by a factor between 1 and 3. Therefore, the SUR will be sensitive to differences in either the PE or the CE, even if the other measure is low. That is, comprehensible sentences with syntactic surprises or syntactically simple sentences that are incomprehensible will be given moderately high ungrammaticality ratings. But, of course, incomprehensible sentences with high predictions errors will be considered the least grammatical.

If human subjects had rated the ungrammaticality of a sentence on a sliding scale, the appropriate measure for comparison would be the SUR. Because subjects would probably be using a closed scale while the SUR is an open-ended measure, some monotonic transformation of the SUR should be used before the actual comparison is made. However, if subjects are not rating the grammaticality of sentences but judging them in yes/no forced-choice decisions, the SUR should not be directly compared to the percentage of "ungrammatical" responses. Rather, the SUR should be converted, on a sentence-by-sentence basis, into a binary decision and the average of many such decisions used for comparison. The reason is that the two measures may not always agree. Two sets of sentences could have similar average grammaticality ratings, but very different average grammaticality judgment scores if the variance of ratings over the individual sentences is different between the two sets.

The simplest way to convert an SUR into an ungrammaticality decision is to pick a fixed criterion and decide that any sentence whose rating is above the criterion is ungrammatical. However, the decisions made by human subjects, especially under pressure, are likely to be a lot more noisy than such a procedure. Therefore, rather than using a fixed criterion, a noisy criterion might be more appropriate. For each decision, the criterion would be set to the sum of a base level and a noise value, which is drawn randomly from a flat distribution centered at 0. The base criterion and the range of the noise distribution could be adjusted to model varying experimental conditions. If the experimental items, including fillers, tend to be mostly grammatical, subjects will probably adopt a low (strict) criterion, while if the items are mostly complex and often ungrammatical a looser criterion would be used. The noisiness of subjects' performance is also likely to vary with the degree to which accuracy or fast responses are encouraged. In the results reported here, grammaticality decisions were simulated using a criterion of 4 ± 2. In order to produce more consistent results with a limited number of sentences, multiple decisions can be made about each sentence, using many randomly chosen criteria. In practice, I have used 100 decisions per sentence. The resulting average will be referred to as the *simulated ungrammaticality decision*, or SUD.

Clearly the methods for computing the SUR and SUD are somewhat arbitrary. The SUR formula, as well as its constants, was chosen to reflect my intuitive sense of the manner in which subjects respond to syntactic and semantic errors in judging grammaticality. However, the current method is flawed in the sense that it makes use of information (the correct answer to comprehension questions) to which the subjects would not normally have access. It is likely that different constants, or a different method altogether, will enable the model's grammaticality judgments to more accurately reflect those of human subjects. But, again, this measure should be a reasonable first attempt.

6.6 Claims and limitations of the model

We have just mentioned a few limitations of the CSCP model and, at this point, most readers will have in mind many more limitations or questionable assumptions, critical or otherwise, implicit in its design. Before beginning the analysis of the model's behavior, it is worthwhile to take a moment and discuss which of these factors are actual claims of the theory underlying the model and which reflect simplifications that were necessary as a matter of practicality.

6.6.1 Claims

The following are some of the major claims of the theory underlying the CSCP model:

- **The human sentence comprehension and production systems can be effectively modeled by a connectionist network.** The major properties of the model, such as its sensitivity to frequency, including semantic factors, its preference for regularity, its ability to resolve ambiguities, its ability to comprehend incrementally, and its limited ability to process complex syntactic structures all emerge directly from general principles of information processing in recurrent connectionist systems. The model cannot profitably be viewed as merely an implementation of a symbolic system in neural network hardware.
- **The comprehension and production systems are highly integrated and rely on common representations.** Not only do comprehension and production both involve the same message representation, the majority of the system is involved in both comprehension and production, with certain aspects more critical for production and others more critical for comprehension.
- **The sentence processing system does not contain clear modules, but it does have functional specialization.** The various components of the model do have functional specializations. The *word input* layer, for example, is not involved in syntactic processing, and the message layer is, by design, mainly concerned with sentence meaning. One could certainly argue that these are modules, but they are modules in a very weak sense. There is a strong degree of functional specialization near the input and output ends of the system where the modality and representational structure of information is highly constrained, largely because of the practical necessities of implementing a model with well-defined interfaces. Where the majority of the work is performed in the model, in the mappings to and from the comprehension gestalt and production gestalt layers, there is weak specialization and a high degree of interactivity. In order to do their jobs, these areas must integrate syntactic, semantic, and lexical information from the current sentence with their general knowledge of each of these three domains, in the form of the learned connection weights. Although one gestalt layer is more responsible for comprehension and the other for prediction and production, one would be hard-pressed to call them distinct modules.
- **In comprehension and production, syntactic processing is not clearly segregated from lexical or semantic information.** To put a finer point on the previous claim, the current model certainly does not involve a separate all-syntactic phase of parsing. Lexical frequency and semantic information pervade all stages of processing. While this is true of many recent theories of sentence processing (MacDonald et al., 1994a; Tabossi et al., 1994; Trueswell & Tanenhaus, 1994), the current model makes the additional claim that the human language system does not necessarily engage in parsing per se. Certainly the model must develop some knowledge of the syntactic structure of the language to do a reasonable job in comprehension and production. But performing these tasks, mapping from a surface form to a message and back again, does not necessarily require the model ever to construct an explicit representation of a sentence's syntactic structure.
- **The comprehension system is constantly engaged in formulating covert predictions.** While the CSCP model comprehends, it also attempts to predict the next word. Aside from its postulated role in learning production, which is discussed next, prediction can also be quite useful in comprehension itself. Prediction provides an additional source of feedback that can help the model learn the abstract structure of the language (Elman, 1991). Simply performing prediction can enable the model to extract many higher level aspects of sentence structure, such as the existence of various word classes including nouns and verbs, the role of closed class words, or the clausal structure of the language. Higher-order representations that develop as a foundation for prediction can also provide a foundation for learning to comprehend, and may be shared by the two interwoven systems.

  A second potential benefit of prediction, although one that is not employed in the current model, is as an aid to word recognition, either in reading or listening. In theory, the more strongly a word is predicted, the more quickly and easily it can be recognized. This is particularly important in noisy environments where the listener may experience severely degraded input that can only be understood with a firm knowledge of the language and a good idea of the possible meaning of the speaker's message.
- **Learning to predict accurately during comprehension, in the presence of varied levels of advance knowledge of the message being conveyed, is a primary mechanism by which we learn to produce sentences.** This is one of the most important claims of the model. Essentially, the argument is that most of the learning that is required for production actually takes place not during production itself but during comprehension. Certainly this does not include the motoric aspects of production, nor does it include the problem of message formulation, but it does pertain to the language-specific middle layer of production—the mapping from message to sentence.
- **Production is regulated through self-monitoring.** Production proceeds through a process of self-monitoring in which newly produced words are fed back into the comprehension end of the system. This leads to a revision

in the message expressed so far and also provides the impetus for the production of the next word. This self-monitoring need not involve physically producing the word, listening to oneself, and recognizing the word again from auditory input. Presumably, the feedback is mainly internal, involving intermediate lexical representations. A similar proposal has been made by Levelt (1993). The feedback may be quite rapid and could presumably arrive at the word input layer in the form of an internal lexical representation before the word has actually been produced. Although the internal representation used for feedback in the CSCP model is phonological in nature, it is possible that the feedback actually occurs at a somewhat higher level and could possibly involve lexical semantic information. It may be possible to tease apart the form of the internal feedback by observing the pattern of speech errors and whether or not they are detected by the speaker, but designing an elegant experiment to test this may be quite difficult.

Importantly, this theory suggests that production is piggy-backed on the comprehension system. Several predictions emerge from this claim. Although certain forms of brain damage may eliminate production with little effect on comprehension aside from perhaps the loss of any benefit from prediction, there should be no way to eliminate comprehension without also severely hindering production. That is, of course, unless the damage occurs below the level at which the feedback enters the comprehension system, but such damage would eliminate comprehension of even isolated words.

Another prediction is that truly simultaneous comprehension and production of different sentences should be extremely difficult as there will be widespread interference. It may be worth noting, however, that this does not preclude the well-attested ability of humans to closely shadow a speaker, or, in other words, to repeat what the speaker is saying with very little delay (Marslen-Wilson, 1973). That is because shadowing is a case of simultaneous comprehension and production of a single sentence. It is actually quite interesting to consider how shadowing might occur in the CSCP model. At any point in the sentence, the comprehension and prediction system attempts to predict the next word given what it has observed so far, but in the absence of any knowledge of the true meaning. If the prediction is a good one, it should help the word recognition system, which has not been included in the model, to more quickly recognize the new word, even based on just the first part of the word. The word production system (also not included in the model) can then output the recognized word even as it is being fed into the comprehension system for another go. The timing of this is very similar to the timing of the near-simultaneous word production and feedback that goes on in regular sentence production in the model.

- **Production proceeds both incrementally and globally.** It may seem from the fact that feedback occurs after every word that production is an incremental process in the CSCP model. However, that is not necessarily the case and that is not a claim of the present theory. There is nothing in the design of the model that precludes it from planning ahead in the representations used by the production gestalt even before it has produced the first word of the sentence. Because the model is forced to prepare each word in a single time step, the model cannot prepare its future productions in a temporal process. But that does not mean it cannot represent the structure of the upcoming sentence, clause, or phrase in a static, distributed pattern of activation. Nevertheless, much of human sentence production is certainly incremental, a fact which I far too often make myself painfully aware. . . of.
- **Learning to comprehend language is largely based on an ability, either learned or innate, to extract meaning from observation of the world.** A necessary assumption of the model is that much of the information conveyed in language can also be deduced through observation of the world and that these observations provide a form of training signal that the child can use to improve its ability to extract meaning from language. For example, a child might see a snarling dog and think, “This doggy isn’t friendly. It might bite me.” At the same time she might hear her mother say, “That dog is mean.” If the child does not know what *mean* means, she might deduce that it has something to do with being unfriendly and/or having sharp teeth. Later on, this can be refined with further observation. An additional source of advance knowledge of sentence meaning may come from adults repeating and rephrasing what they say until the child responds appropriately. The proposal that language learning is aided by the extraction of information from the environment through non-linguistic means is not meant to be a controversial point. All models of language acquisition seem to make a similar claim and, at this point, I do not necessarily wish to endorse one version of this claim over another one.

6.6.2 Limitations

Having stated some of the actual claims of the theory, the following are a number of properties of the model that were not meant to be claims. They merely reflect simplifications needed to reduce the model to a manageable level of complexity.

- Although the model is a simple-recurrent network with one time step per word, that is merely for reasons of efficiency. Conceptually, the theory is really built around the idea of a fully recurrent network that is highly interactive and makes gradual transitions through state space. The SRT reading time measure is meant to reflect the settling time such a system might experience during reading, but it is undoubtedly limited in its ability to do so, and I have already mentioned some of the expected differences between the SRT and measures of self-paced reading time in humans.
- The model does not include the low-level word recognition or production systems, and words are treated as clearly separable wholes. Not only does this gloss over the important problem of word segmentation, it also seems to imply that the word-level recognition and production processes take place in modules isolated from higher-level information. The latter is not a claim of the theory. The real system is expected to have both bottom-up and top-down flow of information throughout. Although word segmentation is an important problem, it seems to be one that can be solved relatively easily through sensitivity to the statistics of the environment, with a little input from semantically and syntactically inspired word predictions.
- The input and output of the comprehension and production systems also lack any prosodic information, including pitch, stress, rate, and pauses. Prosody would be an interesting addition to the model. It is expected that, if provided in the input, prosody should actually be an aid to comprehension. There seems to be good reason to expect that a connectionist model such as this should be able to make use of such information, even if it is weak, inconsistent, or has a complex relationship to the actual message of the sentence. If the model comes to expect clause boundaries to be indicated by pauses or stress and pitch changes, it may experience an increased garden-path effect on subordinate clause ambiguities when prosodic information is denied to it, as during reading. This could allow the model to better match human performance on sentences of this type. In production, the requirement to produce proper inflection would present an interesting challenge for the model. A principal limitation to current computer speech synthesis programs is a failure to produce anything close to natural-sounding inflection. Perhaps neural networks can learn to do this better than rule-based systems, particularly since inflection is largely driven by meaning, not just by syntax.
- The model does not distinguish between listening and reading. The word input representations are based on phonological patterns, rather than orthographic patterns. But the similarity structures among words in the two systems are not that different, so for the time being we can think of the inputs as representing orthography, as long as we are not interested in issues pertaining to irregular spelling. But there are important differences between listening and reading. In listening, the words arrive at a predetermined rate, while a reader must control the flow of information by directing eye movements. A more complete model should distinguish between these tasks, while also addressing their relationship.
- The semantic and CPP systems in the CSCP model are largely separable and share only the message representation. The model even goes so far as to train the semantic system independently of and prior to the CPP system. That the two systems are so distinct in the brain is not meant to be a claim of the model. It is likely that they interact much more broadly than through a single, static encoding of word meaning, and it is also probable that the two systems develop largely simultaneously. Although the semantic system may begin to develop earlier that the CPP system and serve as a foundation for it, the opposite may be true in later stages of learning, with language driving the development of new and more sophisticated semantic concepts. Such mutual training would be an interesting area for future research but is one that is completely glossed over in the current model.
- To be able to specify and test sentence meanings, the method of separating messages into three-part propositions was developed. Although this seems to work quite well for the time being, I do not wish to claim that all messages can necessarily be represented in such a fashion. It is not clear that all messages really have a propositional structure. Ideally, the level of the interface to the model could be pushed back even further and the model could be allowed to develop its own representations of messages in the service of solving various cognitive tasks.
- As mentioned in Section 5.5, the messages derived by the model are rather shallow in nature. The model is not expected to infer implied constituents in a sentence, let alone to perform complex inference. It is not clear to me,

at this point, the extent to which sentences require distinct message representations at different levels of semantic analysis. Is a representation of the literal meaning of a sentence a necessary first step or can the comprehender go straight to high-level inference?

- The current model greatly simplifies the process of learning comprehension by assuming that the correct meaning of a sentence is always available. While there may be times at which this is true, more often than not the listener will not know in advance what the actual message is to be. A more likely scenario is that the listener can predict several possible messages, or parts of messages, perhaps in the form of a collection of reasonable propositions. At a given point in development, the learner will understand much of a sentence he hears, and may only lack knowledge of a few words or phrases. In such a case, the parts of the utterance that are comprehensible can narrow down the set of possible meanings. What remains can serve as a fairly reliable teaching signal for the unfamiliar parts of the sentence.

  As an example, imagine that a child and his mother are in the park and witness the following scene. A squirrel runs out of the bushes with a big dog hot on its heels. The dog crashes into a baby carriage, knocking it over. The squirrel runs up a tree with bright red leaves and the dog starts barking at the squirrel. There are all sorts of possible messages the child might expect here: "The dog is running," "The squirrel went up a tree," "The tree has bright red leaves." Predicting in advance what the mother might say is very difficult. But if the mother says, "Look, the dog *blarp*ed the squirrel," and the child knows about dogs and squirrels, and has picked up on the SVO word order of English and the fact that stuff ending in "-ed" already happened, it might guess that *blarp* means *to run after*. It could not mean *to bark at* because that is still going on. If the mother says, "Look, the dog knocked over the *floopalump*," and the child has an idea what knocking over is, it might learn that a *floopalump* is a basket with wheels for babies.

  Thus, relying on observation of the world as a training signal for language learning may be a multi-stage process. A difficult question for future research is how learning takes place in such a situation. Is the sentence processed twice—once to obtain an approximate meaning and to narrow down the set of possible interpretations and again to learn to comprehend it better? Is the sentence stored in its entirety and replayed internally, or is a representation formed of just those parts that did not make sense to serve as a basis for learning? The failure to address these questions is perhaps one of the most critical limitations of the current model as an account of development.

- A related issue is the fact that the model learns all words in the context of processing sentences. Children, on the other hand, seem to be able to learn the meanings of concrete nouns, physical actions, descriptive adjectives, or proper names through individual pairings of words with visual inputs, and to then productively use those words in sentences. Many generalization experiments rely on this ability. Older children and adults can also learn new words by being told their definitions. It is possible that such abilities rely on a distinct set of processes that are not involved in comprehension and production of familiar words and also rely on links between the sentence processing system and the world model or other forms of short-term memory in which new concepts can be quickly formed and associated with new words.

- Discourses in Penglish are limited to single sentences. There is little sense of discourse coherence, with the exception of some clausal relations within a sentence. A corollary of this is that all sentences occur in a vacuum. There is no provision for a discourse model or world model, both of which are believed to be important components of the human language system, or, perhaps, of the cognitive system in general. True comprehension involves the problem of updating and removing information in one or more models of the world, including both real facts and those pertaining to hypothetical situations.

- Penglish noun phrases, like those in many of our empirical tests of sentence processing, have the peculiar property that they all introduce new actors or new entities. Understanding such sentences involves building new instantiations of each of these constituents. Real language, however, is fraught with anaphoric and other forms of reference. Only some noun phrases introduce novel constituents. Many, if not most, refer to things that have already been introduced in the discourse. Comprehending such a sentence requires not just instantiating noun phrases but resolving references and recalling elements from the discourse or world model. This is not expected to be difficult, in principle, for a connectionist system, but its inclusion would greatly complicate the current model. Nevertheless, the addition of reference should be one of the first extensions of the CSCP model.

- The model is unable to perform reanalysis. It must process a sentence sequentially and cannot return to earlier material. A more complete system should be able to self-monitor its progress and, when it gets lost, jump back to an appropriate point. This could be done visually in the case of normal reading. During listening or during masked reading, this must be done with some sort of short term, perhaps phonological, memory buffer. Reanalysis of this sort is actually quite complex as it requires the ability to discard those aspects of the message that derived from the incorrectly processed portion of the sentence. Thus, the comprehender must maintain some sort of connection between the surface form of the sentence and the meaning that has been extracted from it.

- Presently, the model does not solve the process of message choice during production, but is always provided with the message that it must produce. If message choice and sentence production were encapsulated processes, this would not be such a serious limitation. However, as anyone who has tried to learn a foreign language knows, message choice often depends on the abilities of the production system to convey that message. With non-fluent adult speakers and with children, message choice and sentence production are bound to be interacting processes. Any reasonable account of language production during development must provide an explanation for both message formation and message expression.
- Finally, although the model may account for some important aspects of language development at a higher level, the simple-recurrent backpropagation-through-time learning method used in the CSCP model does not seem to be a reasonable approximation to the mechanisms of learning in the brain. In addition to the backward flow of information in standard backpropagation, BPTT requires storing and reactivating past activation states in reverse temporal order. Presumably the language systems of the brain actually use some form of Hebbian learning. However, our current understanding of Hebbian learning is basically limited to learning over a single set of weights. In order to provide a more reasonable account of learning in the brain, we will need to investigate variants of Hebbian learning that are capable of operating in multi-layer networks and in recurrent networks processing temporally varying signals.

Chapter 7

General Comprehension Results

The previous chapter explained the operation of the CSCP model, as well as some of the theory behind its design. This chapter begins the analysis of the behavior of the model with a number of experiments evaluating the model's comprehension and reading time performance. It should provide a general idea of the competency of the model in processing the Penglish language, as well as introduce some factors that govern the model's behavior. More detailed experiments pertaining to the particular ambiguities or structures discussed in Chapter 3 are described in the succeeding chapters.

7.1 Overall performance

As described in Section 6.4, three instantiations of the model, Adam, Bert, and Chad, were trained. For each network, the semantic encoder system was trained for 16 epochs of 250,000 sentences and then the CPP system was trained for the same 16 epochs. The semantic encoder stores all of the propositions in a sentence into a static message. Its ability to do this can be evaluated by asking the decoder to answer all possible questions about those propositions by relying on just the information stored in the message. A "question" is in the form of a proposition with one of the three parts missing, and the model must respond to the question by filling in the missing part. As explained in Section 6.5.1, question answering performance can be measured in two ways. The *strict* criterion requires that the activation of all proposition response units be on the correct side of 0.5 for the answer to be considered correct. The *multiple-choice* criterion is somewhat more lax. It requires that the part of the proposition provided in response to the question be closer to the correct answer than it is to any of four other distractors, which are of the same semantic class as the answer. If the answer is an object, the distractors will be other objects. When possible, the distractors are drawn from other items in the same sentence.

The performance of the encoder system over the course of its training is shown in Figure 7.1. At each point, the model was tested on the first 5,000 sentences in the testing corpus. Because the testing corpus is representative of Penglish as a whole, the distribution of propositions per sentence in the sample should approximate that shown in Figure 5.3. Figure 7.1 displays the performance of the three different networks using separate lines. However, because the error rates of the networks are so similar, it is not practical to distinguish between them. In this and most subsequent figures, error, rather than correct performance, is plotted. Therefore, lower is better. During its first epoch of training, performance improves rapidly, reaching 38% error with the strict criterion and about 11.6% error with multiple-choice. By the end of training, the average encoder error reaches 11.8% with the strict criterion and 2.9% error with multiple-choice.

Figure 7.1 does not show the comprehension performance of the network, only its ability to encode and decode the propositions in a sentence. The comprehension performance is shown in Figure 7.2. In this case, the networks must listen to (or read) the words in the sentence and, from them, extract a message. The message can then be queried using the same method as for the semantic encoder. Ideally, the comprehension system should be able to produce the same message that results from encoding. In practice, there will be some mismatch between the two messages, and question answering performance is likely to be worse following comprehension. Once again, results from all three networks are

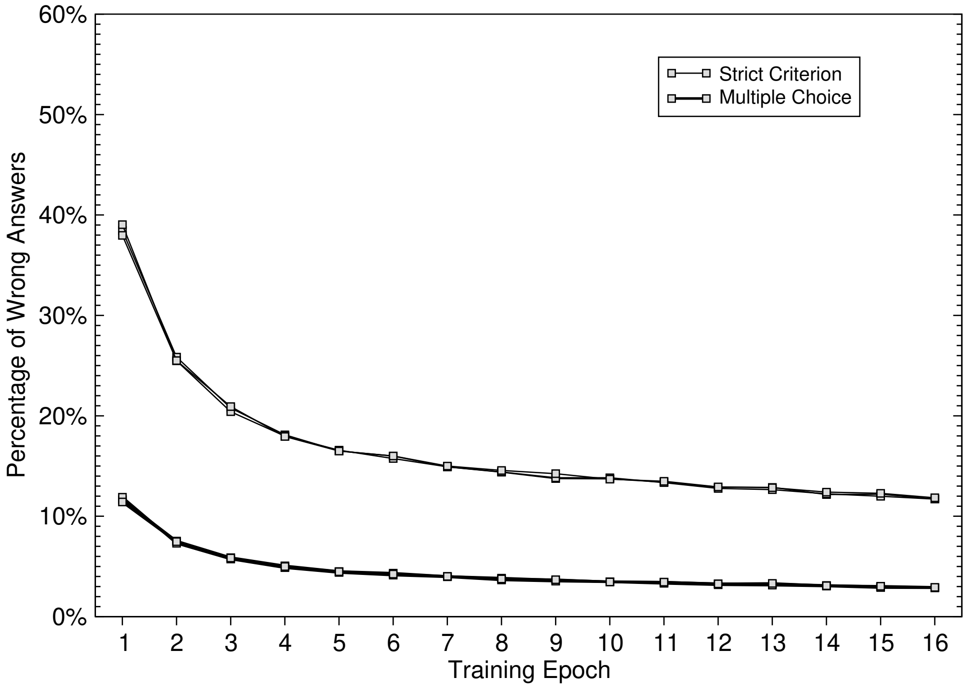


Figure 7.1: Encoding performance of the semantic systems during training.

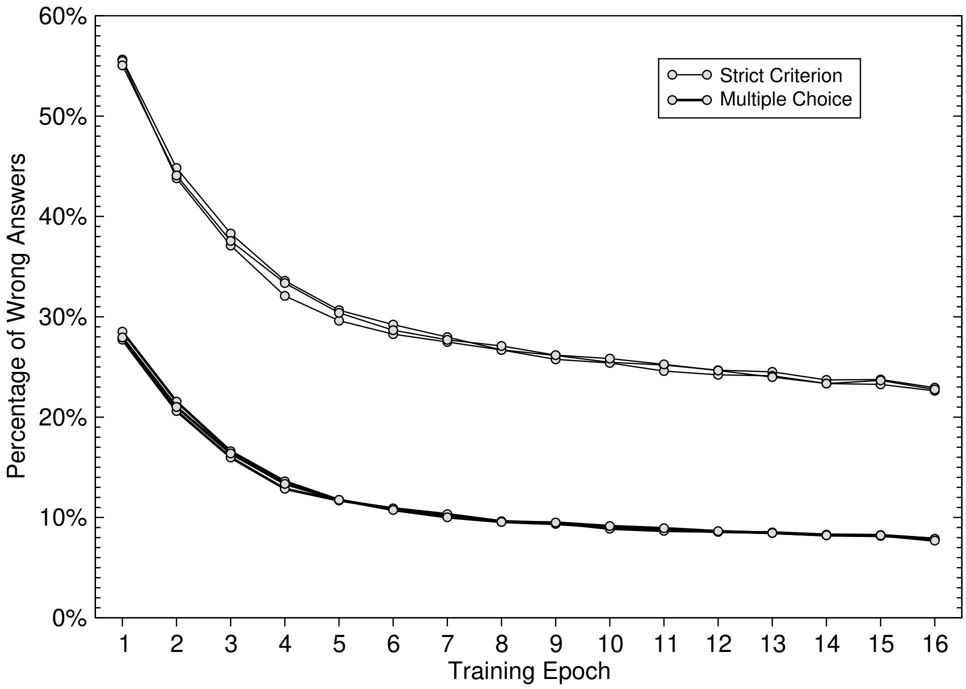


Figure 7.2: Comprehension performance of the full network during training.

shown, but there is little difference in their performance. By the strict criterion, the networks reach 55.4% error after one epoch and 22.8% error after 16 epochs. Using the multiple-choice criterion, the networks reach 28.1% error after 1 epoch and 7.8% error by the end of training. Thus, averaging over sentences in the testing corpus, the final networks are able to provide the complete, accurate answer to 77% of questions. If given five-choices, they can pick the correct one 92% of the time.

Of course, averaged measures such as this do not tell us anything about the types of errors made or the ability of the networks to handle complex sentences. Figures 7.3 and 7.4 show the performance of the networks as a function of the number of propositions in the sentences, using either the strict or multiple-choice criteria, respectively. As a point of reference, the following are some examples of sentences whose meanings contain from 1 to 8 propositions:

1. The teacher has read.
   The question was asked.
2. Some lawyers knew a story.
   The manager had been questioned by the owner.
3. The girl put the table on the floor.
   The lawyer read me a question.
4. You will play baseball because owners were seen.
   The boy believes that the teachers took me.
5. Because that plane was taken the father said that the girl killed.
   Mothers that bought that table bought the evidence of the book.
6. A father knows the boy hoped the player throws something.
   The manager thinks the park is left but you play with us.
7. The cop knows that a nice mother on a big plane was asked for the boy.
   I realized that the manager will see that something was heard by a nice boy.
8. The lawyer that the dog saw realizes managers believed the balls were thrown to the house.
   Managers told the owner threw the apple but the mother will leave the picture in the park.

As should be expected, performance generally gets worse as the number of propositions increases. For the message encoder, this is true monotonically, with error rates ranging from 1.3% for one-proposition (1-prop) sentences to 28.5% for 8-prop sentences. However, using the strict criterion, the error rate for the complete comprehension system does not increase monotonically with number of propositions. 1-prop sentences are harder than 2-prop and even 3-prop sentences. One thought is that this might be due to the nature of the propositions in those sentences. The proposition in a 1-prop sentence always describes the relationship between the subject and verb, which could either be an agent or experiencer relationship. Object/action relationships tend to be the hardest ones because objects and actions tend to have more variability in their descriptive features than do properties (adjectives or adverbs). Because many 2-prop sentences contain one proposition with an object/property or action/property relationship, one might expect the average difficulty of the questions in 2-prop sentences to be somewhat easier than those in 1-prop sentences. However, it turns out that this is not an adequate explanation for the fact that 1-prop sentences are so difficult. If it were, the difference should be seen in the encoder as well as in the comprehender. The real explanation has to do with an unexpected difficulty arising in intransitive sentences, which will be explained shortly.

On easier sentences, having from 1 to 4 propositions, the message encoder performs quite well. The majority of error is due to the comprehension system. On more complex sentences, the performance of the encoder rapidly declines. With 8-prop sentences, the strict-criterion error rate of the complete comprehension system is only about 50% higher than the encoding error rate. Thus, the majority of errors may be attributable to the representational limits of the semantic system, rather than to sentence comprehension itself. As seen in a comparison of Figures 7.3 and 7.4, the error rate using the multiple-choice criterion is much lower than that with the strict criterion. The multiple-choice error rate is also very nearly monotonic for both the encoder and comprehender, without the elevated error rate for 1-prop sentences. This is largely because 1-prop sentences contain no useful distractors. Therefore, all distractors must be drawn at random and are thus not very distracting.

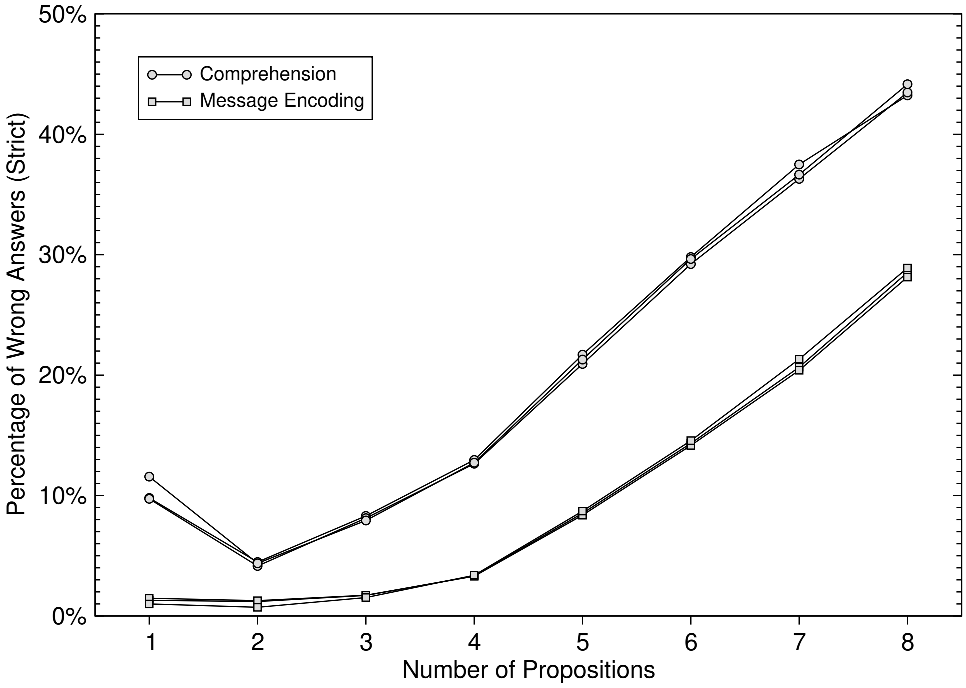


Figure 7.3: Strict-criterion performance of the trained model based on the propositions per sentence.

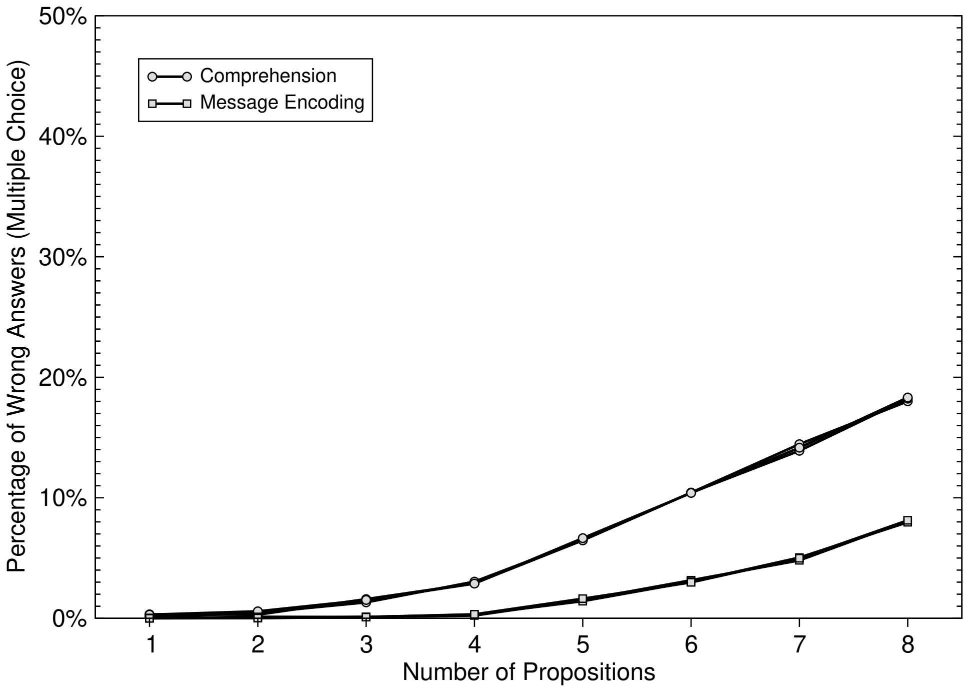


Figure 7.4: Multiple-choice performance of the trained model based on the propositions per sentence.

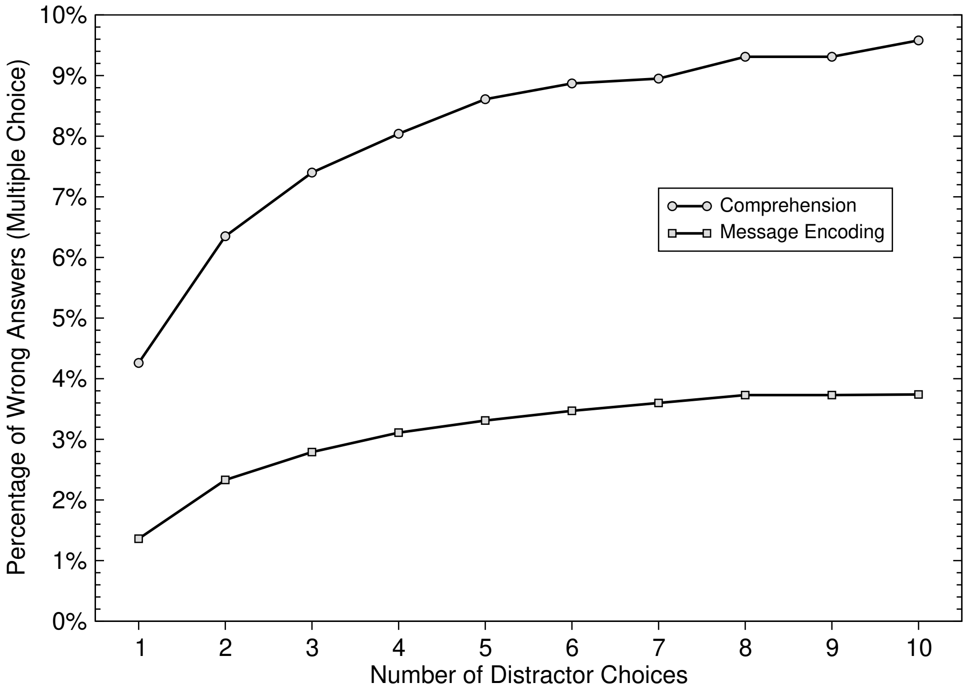


Figure 7.5: Multiple-choice performance of Adam as a function of the number of distractors.

Although the model has trouble with very complex sentences and tends to make a few comprehension errors per sentence, it is still able to understand some fairly long sentences perfectly. The following are some of the 6-prop sentences that are comprehended by Adam with 100% accuracy using the strict criterion:

- Players have bought a book and the father went to a school quickly.
- I think girls are using evidence in the small house.
- The manager had had new pictures I bought in a school.
- The car had been had in the small house yesterday and I took a manager.

When using the multiple-choice criterion, one interesting question is how sensitive the error rate is to the number of available choices. Would performance decline significantly if a few more distractors were added? Figure 7.5 shows the relationship between the number of distractors and multiple-choice error rate for both message encoding and comprehension. The results are based on evaluating the Adam network on the first 1,000 testing set sentences. Clearly, the error rate increases only gradually with an increased number of distractors, with the greatest changes occurring between 1 and 5 distractors. This is largely because distractors are first drawn from the other constituents in the sentence that are of the same semantic type as the answer, and thus might be confused with it. The remaining distractors are selected at random from constituents (of the same type) used elsewhere in the language. These out-of-sentence constituents are rarely selected by the network. When it is confused, it tends to produce a response that is a mixture of the correct answer and some other constituent that appeared elsewhere in the sentence. This tendency to be distracted by other constituents in the sentence is presumably similar to the behavior that humans would show on a similar multiple-choice task. All other experiments involving the multiple-choice measure are conducted using four distractor items.

7.1.1 Experiment 1: Generalization to novel sentences

Another important question is how well the model is able to generalize to novel sentences. Have the networks simply memorized the sentences in the training sets or have they learned general properties of the language's syntax and

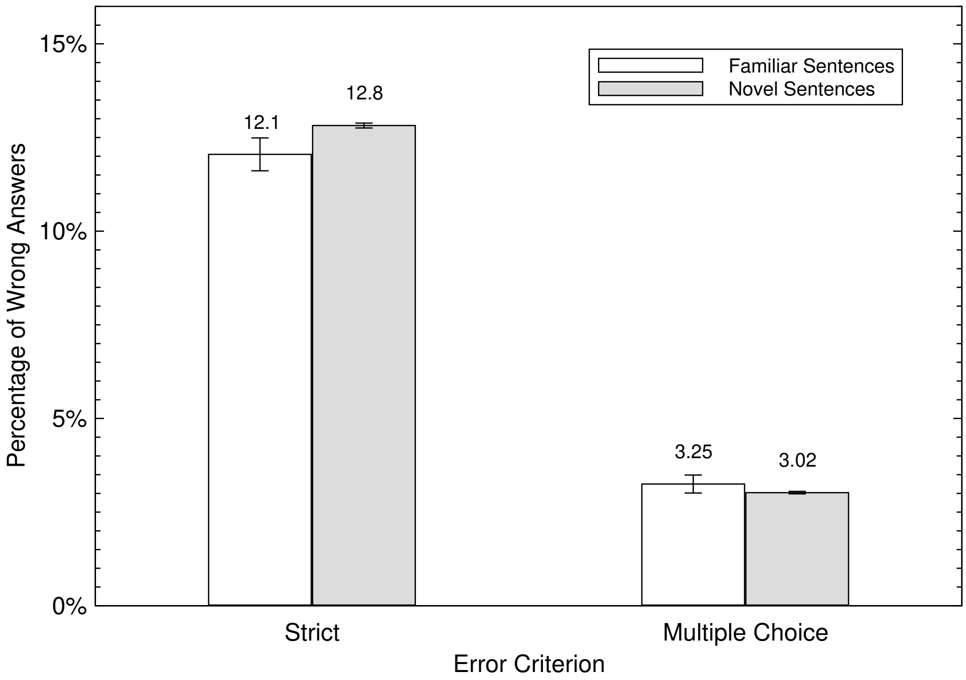


Figure 7.6: Comprehension performance on familiar and novel 4-proposition sentences.

lexicon that should enable them to comprehend and produce novel sentences. One way to test the generalization ability of the network is simply to compare its performance on testing set sentences that did or did not also appear in the training set. However, a potential problem with this approach is that it may introduce bias into the two sets which could affect the results. Sentences that appear in both the training and testing set tend to have higher overall frequency in the language. They will probably contain higher frequency syntactic structures, higher frequency words, and higher frequency semantic relations. In contrast, the novel sentences may never have occurred before specifically because they contain abnormalities. If the model proves to be better at the familiar sentences, will it be because they are familiar or because they are composed of higher frequency components or simpler constructions?

The potential for such bias is particularly strong for short sentences. But as the complexity of the sentences increases, so too does the proportion of novel sentences in the testing set. As was shown in Figure 6.5, only 2% of the 4-prop sentences in the testing set are familiar to the model. With such a low proportion of familiar sentences, we should expect less sampling bias in separating the familiar from the novel sentences. Therefore, a reasonable comparison between the two sets can be made. Figure 7.6 shows the comprehension performance on the 153 familiar and 7,482 novel 4-prop sentences, averaged across the three networks. Using the strict criterion, comprehension is slightly better on the familiar sentences, but not significantly so.[1] But according to the multiple-choice measure, performance is numerically better on the novel sentences, although again not significantly.[2] Therefore, there is no indication that the model is significantly worse on novel 4-prop sentences. It is possible that a significant difference could emerge if more familiar items or more networks were available to be tested, but it is unlikely that any difference would be a substantial one. The model is clearly capable of a good deal of generalization and is certainly not operating on the basis of having memorized the individual training sentences.

7.2 Representation

These next few analyses focus on the message representations, which are developed by the network in learning to encode and decode proposition sets and which are shared by the comprehension and production systems. The en-

[1] F(1,273334)=2.36, p=0.092
[2] F(1,273334)=0.928, p=0.335

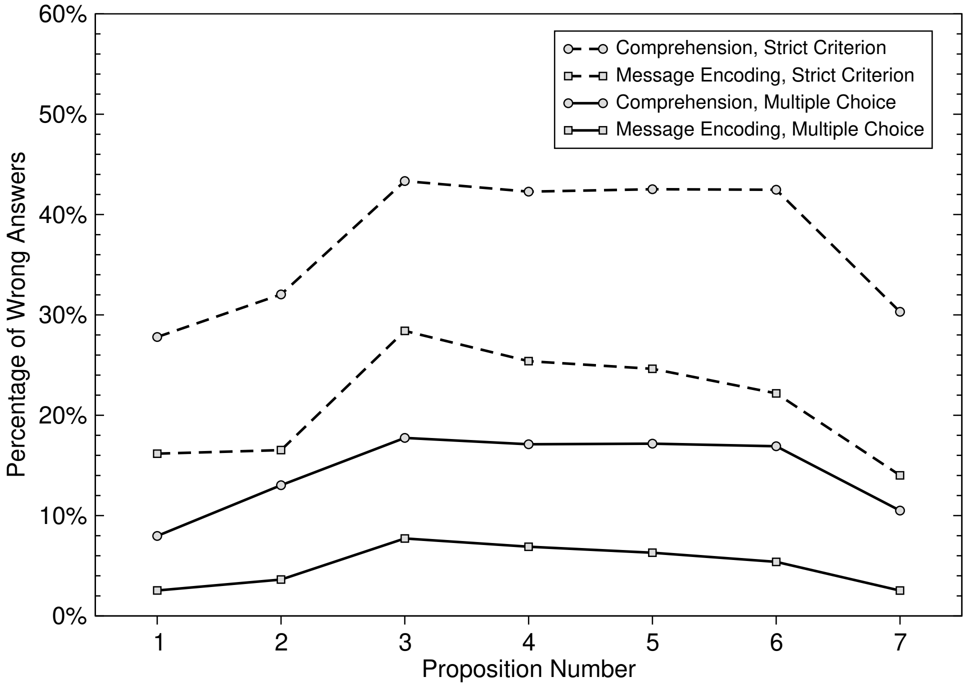


Figure 7.7: Question answering as a function of the order in which propositions are stored in the message.

coder system builds the message representation by storing one proposition at a time. Earlier, we speculated that, in recalling or answering questions about these propositions, the model might show primacy and recency effects, as are commonly found in human memory research (Glanzer & Cunitz, 1966). This is confirmed by plotting question answering performance as a function of the order in which the queried proposition was stored in the message, as shown in Figure 7.7.

All of the data in the figure is from the encoding and comprehension of 7-prop sentences, although similar results are obtained with any sentences having more than two propositions. The two lower curves reflect multiple-choice error while the upper curves reflect the strict-criterion error. Performance of both the encoder itself and the comprehension system are shown. All four curves have the classic U-shape, although the U's are inverted because error is being plotted. Each of the curves is quite low for the first two propositions, jumps to a maximum on the third proposition, gradually falls for the next three propositions, and then jumps down on the last one. Therefore, it seems that the effects of primacy are limited to the first two propositions, while those of recency are mainly concentrated on the last proposition, although they do seem to extend back to the fourth proposition. It is interesting that similar effects are seen in both encoding and comprehension. However, there does seem to be a subtle difference between the two. The encoder curves gradually decline from 4 to 6 propositions, while those for the comprehender remain flat. This suggests that the more subtle effects of recency are lost when messages are composed through comprehension, rather than through direct encoding.

Another way to analyze the error rate of the model is by the part of the proposition being queried. Each probe presents a partial proposition, with one of the three parts missing, and the correct response is the complete proposition with the missing part filled in. So one might ask which of the parts tend to be the hardest to fill in. The answer is given in Figure 7.8. Responses are shown following either encoding or comprehension and measured with either multiple-choice or the strict criterion. In all cases, filling in the middle part of the proposition is the easiest, with the right part the hardest and the left part slightly easier than the right.

One factor that might contribute to the middle part being the easiest according to the strict criterion is that it contains only 25 bits while the left and right each contain 134 bits. However, this does not necessarily explain why it is also the easiest according to multiple-choice. The middle part of the proposition encodes the relationship between the left and the right. Sometimes this relationship is easy to guess just based on the types of the left and right components. For example, if the right component is a property, the relationship must be property. But in the majority of cases, the

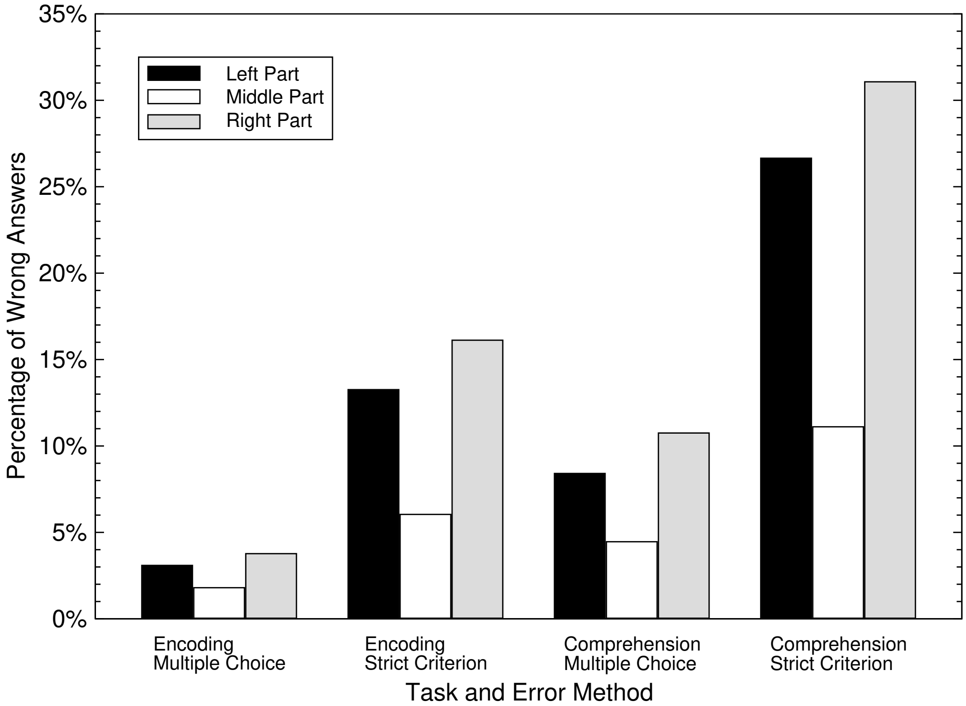


Figure 7.8: Question answering as a function of the part of the proposition that must be filled in.

middle part of the proposition represents either a thematic role or a relationship between two actions, which cannot usually be deduced from the other parts themselves. Sometimes the difference in thematic roles can be quite tricky. For example, the subjects of some verbs are agents, while others are experiencers, and some objects are patients while others are themes. The encodings of each of the different thematic roles differ in just two bits. Therefore, on a multiple-choice test the distractors are, on average, not far from the correct answer and any mistake might well result in choosing a distractor. In contrast, an action or object and its distractors will tend to differ in more than two bits. This may explain why the drop in performance between the strict and multiple-choice criteria is smaller for the middle part than it is for the outer parts.

Filling in the outer parts of the proposition are harder as they tend to be actions and objects that have more features and which cannot be deduced from the other members of the proposition. Objects, furthermore, tend to be harder as there are generally more nouns in a sentence than there are verbs, resulting in more attractive distractors, whether or not the multiple-choice criterion is used.

Turning to a very different issue, we might ask an important set of questions about the CSCP model regarding the way in which information is encoded in the message representations. The encoder is tasked with storing information in the form of a high-dimensional vector. But how is that vector structured? Are the unit activations relatively stable as new information arrives? Do they flip back and forth between positive and negative phases on every step? How is old information altered to make room for the new information? A priori, without much experience with encoder networks of this kind, it is difficult to predict the message encoding behavior of the model. Because the message units are able to completely update their states on every step, there is no mechanism directly forcing them to change their activation gradually. Likewise, there is no mechanism encouraging the message representation to be sparse.

Nevertheless, it happens to be the case that, during training, the networks consistently develop rather sparse message representations that are quite stable over time. As more propositions are stored in a message, the overall density of active units increases. Figure 7.9 shows four different measures of the message representation during encoding. “Average Activation per Unit” is the average activation, or output strength, of the message units. “Total Units On” is a related measure that indicates the percentage of units whose activations are above 0.5. In general, only about 10% of the units are active at any one time. Furthermore, the average activation and the number of active units increases as more propositions are stored in the message array. This indicates that information in the message representation is

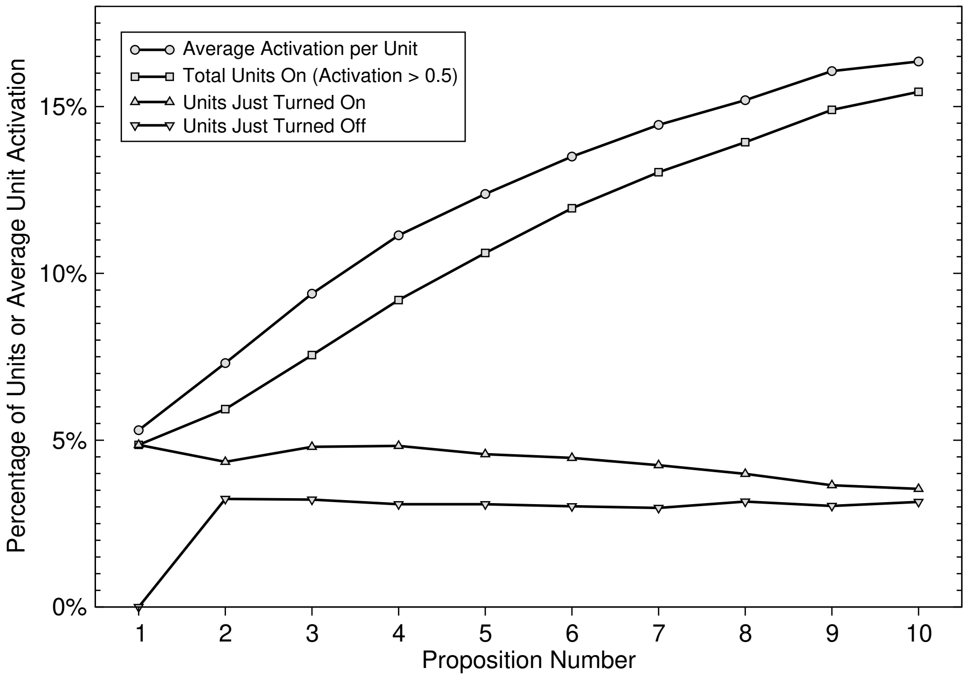


Figure 7.9: Average activation, active units, and changes in unit activity in the message layer during message encoding.

basically additive. There is a positive relationship between overall activation and the amount of information stored in the message.

One might ask whether activation in the message representation strictly grows over time, or whether there are more complex dynamics to it. To help answer this question, the average changes in active units with each new proposition are also plotted in Figure 7.9. "Units Just Turned On" is the percentage of units that were activated by the arrival of the most recent proposition. That is, the units whose activation went from below 0.5 to above 0.5. "Units Just Turned Off" is the percentage of units that were deactivated when the last proposition arrived. Integrating the difference between these two over the proposition number should give the total units on. If the message were becoming strictly more active with added propositions, we would expect very few units ever to turn off. It turns out, however, that units are being turned both on and off, although more tend to be turned on than off as propositions are added.

Figure 7.10 shows the same four measures, but, rather than being plotted as a function of proposition number during encoding, they are plotted as a function of word number during comprehension. In this case, as one might expect, changes are more gradual, since each word provides less information than each proposition. In the case of comprehension, there is a greater difference between the percentage of units with activation over 0.5 and the average unit activation. This means that, during comprehension, more of the total activation is carried by units that are only weakly active than it is during encoding. One could speculate that this is due to the need to represent multiple possible interpretations as a result of ambiguity during comprehension.

Clearly this is only a first step of an attempt to understand how information is represented by the CSCP model during encoding and comprehension. Many, more in-depth, experiments are possible to tease apart how specific types of information are represented, particularly during the comprehension of locally ambiguous sentences. The fact that the message representations are fairly stable and tend to increase in activation as new information arrives is very useful, as it suggests that the message representations should be fairly open to analysis. Rapidly changing representations would be quite difficult to break down into meaningful components. Another benefit of the type of representations developed by the model is that activation change is a potentially reliable measure for information change. Therefore, activation change in the message layer can be a useful component of an analog to human reading time. The reading time measure is discussed further in Section 7.7.

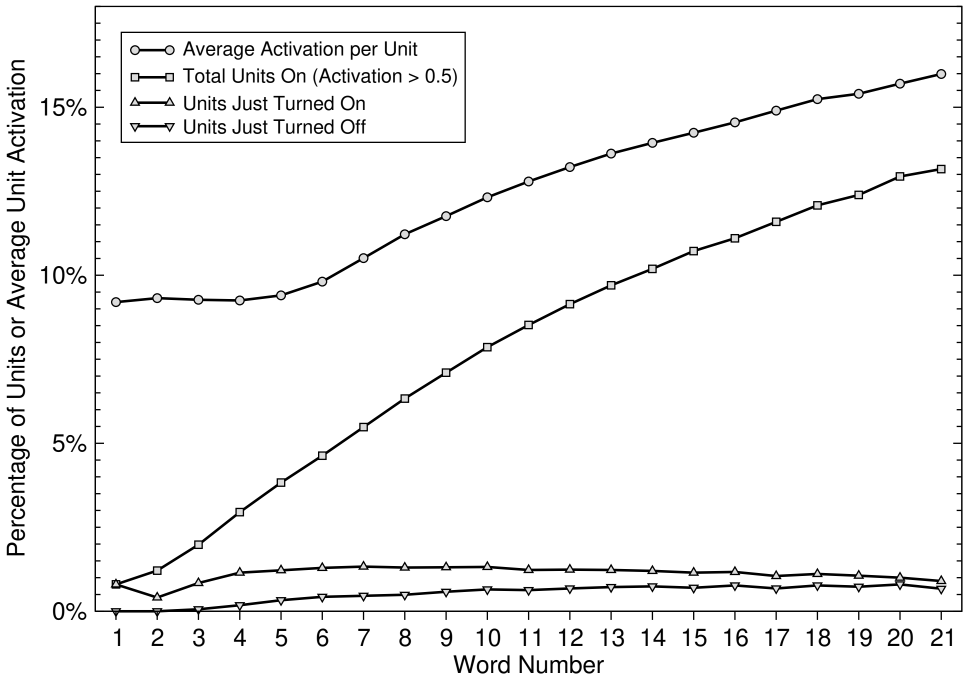


Figure 7.10: Average activation, active units, and changes in unit activity in the message layer during sentence comprehension.

7.3 Experiment 2: Comparison of sentence types

To this point, we have only looked at general measures of comprehension, averaged across many different sentence types. We now look at how the model performs in comprehending sentences with specific syntactic forms. Table 7.1 shows 25 different sentence types and lists the number of propositions in the meaning of each type and the frequency, out of a million Penglish sentences, with which sentences of that type occur. Minimal differences between classes of simple sentences can be used to help understand the ease with which the model can handle various structures. The sentence types here are defined by their clausal structure, voicing, and the presence of prepositional phrases and adjectives. The definiteness or number of a noun phrase and the tense or aspect of the verbs do not distinguish different sentence types. Together these sentence types account for about 45% of all sentences in Penglish.

Figures 7.11 and 7.12 show the error rate of the model on each of the 25 sentence types, using either the strict criterion or multiple-choice, respectively. Two hundred sentences of each type were tested using all three networks. The easiest sentences for the network are the simple transitives, TRNS, which result in only 2.2% error by the strict criterion. This is both because they are so simple, containing only two propositions, and also because they are so common, accounting for 12.7% of all sentences. One might expect that, because basic intransitive sentences, INTR, contain only one proposition, the model would find them even easier than transitives. However, the model clearly has considerable difficulty with intransitives, with 17.5% error by the strict criterion.

Why are intransitives so hard for the model? It does not seem to be a matter of their low frequency. Intransitives in Penglish are at least half as common as transitives. Their somewhat lower frequency might, at best, account for a small difference in comprehensibility. The answer seems to be in the representational differences in the meaning of transitives and intransitives, in conjunction with the operational limitations placed on the model and the fact that most intransitive verbs in Penglish also use transitive argument structures. As was shown in Table 5.2, Penglish has very few obligatorily intransitive verbs. The majority of verbs are at least optionally transitive, and of the 25 verbs that can be either intransitive or transitive, 16 are more common as transitives. Therefore, during comprehension, when the model has read, "The mother saw...," it will most likely be expecting a transitive sense. In such a situation, the model will tend to form an interpretation that is biased toward a transitive reading, and which may be far, in semantic space, from the correct intransitive interpretation. When the next word arrives, and turns out to be an end-of-sentence

Type	Props	Frequency	Example Sentence
INTR	1	66,480	The mother saw.
PASS	1	93,713	The plane has been found.
INTR-ADVB	2	6,065	A teacher will ask tomorrow.
INTR-VPP	2	19,160	The dogs went to the house.
INTR-ADJ	2	485	The big car is leaving.
TRNS	2	126,549	The manager found the owner.
PASS-BY	2	7,701	Something is seen by the mother.
DITRNS	3	9,986	The lawyer will give teachers books.
DITRNS-P	3	1,408	A cop has taken something to the owners.
TRNS-SADJ	3	980	The young girl found a bat.
TRNS-OADJ	3	27,699	A player drove a small car.
TRNS-SNPP	3	7,586	The manager of a park wants a picture.
TRNS-ONPP	3	31,627	The father saw the birds on the house.
TRNS-VPP	3	25,827	Girls had written tests in school.
SC	4	20,690	The boy knows the dog will follow a girl.
TRNS-ADJ	4	252	The loud cop bought the small picture.
TRNS-NPP	4	1,959	A mother of the girl followed the mother of a boy.
CONJ-TI	4	1,426	The boy said something and the reporters left.
CONJ-IT	4	1,441	The boy had hoped but the girl played baseball.
SUB-R-TI	4	379	The girls will get you because cops had left.
SUB-R-IT	4	369	I asked because managers used the baseball.
SUB-L-TI	4	541	If I have bought a car something is flying.
SUB-L-IT	4	569	Although girls told a father will give something.
RC-CS	4	601	The teacher that has a car followed me.
RC-CO	4	71	Teachers who the managers had seen follow me.

Table 7.1: 25 sentence types discussed in this section.

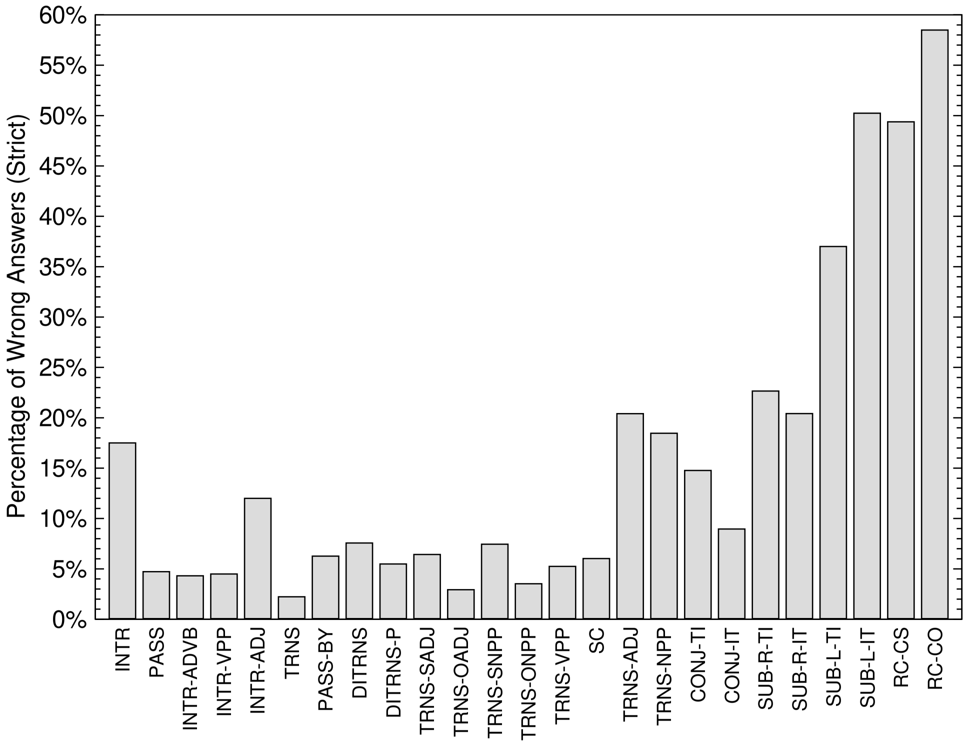


Figure 7.11: Comprehension performance on the various sentence types using the strict criterion.

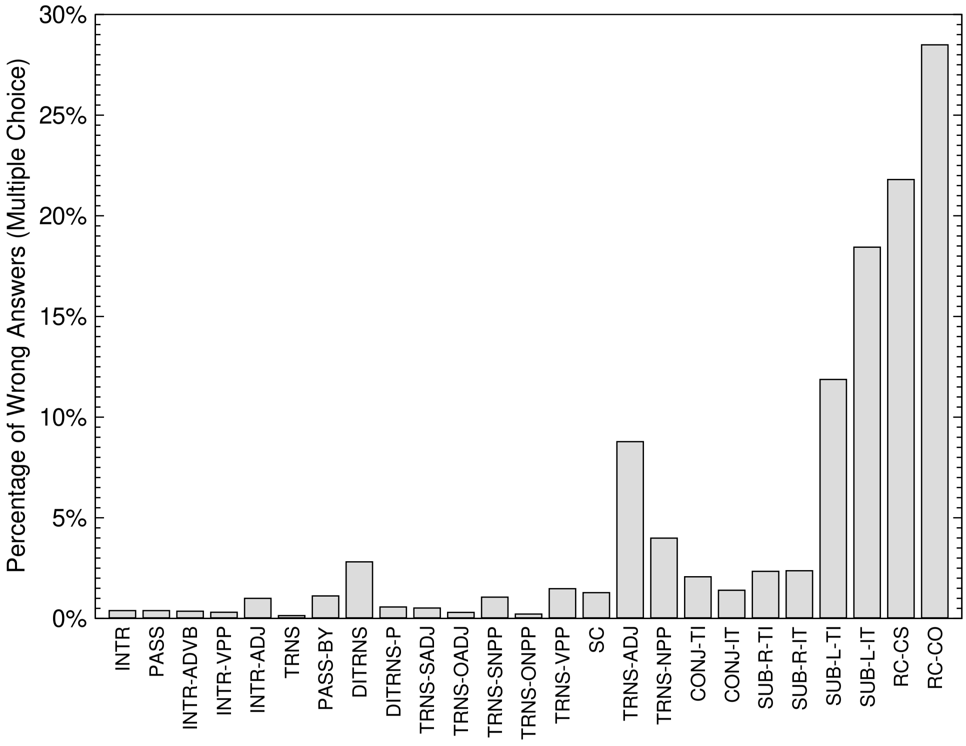


Figure 7.12: Comprehension performance on the various sentence types using the multiple-choice criterion.

marker, the model must immediately revise and finalize its interpretation. The end of the sentence is the point of disambiguation. It seems that one time step is just not enough for the model to switch from a transitive reading to an intransitive reading. I expect that, if the model were given an extra step at the end of each sentence to finalize its interpretation, the difficulties with intransitives would mostly disappear.

This explanation is supported by a few additional pieces of evidence. The verb *bark*, which is unambiguously intransitive, does not cause major comprehension difficulty. The strict error rate on INTR sentences using bark is just 4.7%. More convincingly, intransitives actually become easier when they are followed by additional modifiers. An INTR-ADVB is an intransitive in which the verb is post-modified by an adverb, while an INTR-VPP has a verb-modifying prepositional phrase. Despite the fact that they are semantically more complex and less frequent, both of these sentence types are comprehended more easily than are bare intransitives. Presumably these sentences are easier because, while processing the additional words after the verb, the model has time to revise its initial transitive interpretation. As we might expect, when the subject of the intransitive is pre-modified by an adjective, as in an INTR-ADJ, the sentences do not become much easier because the problem of early termination remains. The reason that the error rate of INTR-ADJs is somewhat lower than that of INTRs is that answering questions about the relationship between the adjective and the subject is quite easy, bringing down the average.

Finally, the model is actually much better on simple passives, PASS, than it is on intransitives. Presumably this is because the passives are only temporarily ambiguous, while the auxiliary verb is being processed. By the time the main verb arrives, the construction is clearly a passive. Furthermore, a passive verb is followed by a by-phrase expressing the agent less than 10% of the time. Therefore, the model is not facing a strong bias to expect a by-phrase, as it would expect an object after an optional intransitive.

When a by-phrase does occur after a passive, as in a PASS-BY, the model is reasonably good at comprehending the sentence, but not as good as it is with actives. The strict error rate of a passive with both agent and patient specified is 6.3%, or a bit under three times the error rate of a simple active sentence. This difference is to be expected, and may be due to several factors. Passives, particularly those with by-phrases, are less frequent than actives. Passives also tend to be longer, averaging two words more than TRNS sentences. Finally, passives use non-canonical word order, with the agent or experiencer appearing after the verb. Nevertheless, the model still performs quite well on passives.

The next set of interesting sentence types is the ditransitives. A DITRANS is a double object dative, while a DITRNS-P is a prepositional dative. Despite the fact that they are much less common and use more words, prepositional datives are significantly easier for the model, using either the strict[3] or multiple-choice criteria. This is most likely attributable to the fact that double object datives are temporarily ambiguous. Unless there are strong semantic constraints against it, the first object is probably initially treated as a direct object. When the second NP, which is actually the direct object, arrives, the interpretation must be revised. That the model may have trouble with this revision is supported by the fact that the multiple-choice error rate is disproportionately high for DITRANS sentences. The most likely explanation for this is that the model is confusing the direct and indirect objects, which are included in each other's distractor sets.

Some ditransitives, however, do not have ambiguous direct and indirect objects. *Throw*, for example, must have a ball or other small object for a direct object, but must have a human or a dog for an indirect object. Thus, if *throw* is followed by a human or dog, that argument could not be the direct object, and must be the indirect object, or recipient. Other unambiguous ditransitives in Penglish are *buy*, *read*, *write*, and *tell*. Note that the unambiguous ditransitives are not unambiguous in the sense that they never occur as single object transitives. In fact, they all occur overwhelmingly more often as simple transitives. They are unambiguous in the sense that their direct and indirect objects cannot, semantically, be confused. When ditransitives are partitioned into those with ambiguous arguments and those with unambiguous arguments,[4] the unambiguous ditransitives are significantly easier according to both the strict criterion (11.34% vs. 18.67% errors)[5] and multiple-choice (1.34% versus 10.94%).[6] This is despite the fact that the ambiguous ditransitives are over 50% more frequent, as a class. Therefore, in the case of ditransitives, the model is able to rely on semantic properties of the constituents to help resolve temporary syntactic ambiguities.

Moving on to other sentence types, we consider some other variants of simple transitives. Modification of the object by an adjective, as in a TRNS-OADJ, causes little change in difficulty. By and large, adjectives are not very

[3] $F(1,10600)=18.87$, $p<0.001$

[4] In partitioning the ditransitives, those using the verb *give* were removed because it is only partially ambiguous and it is overwhelmingly frequent as a ditransitive and thus quite easy for the model.

[5] $F(1,10660)=113.6$, $p<0.001$

[6] $F(1,10660)=446.3$, $p<0.001$

difficult for the model. This should not be too surprising since adjectives in Penglish are syntactically very simple, always occurring just before the noun. When the adjective modifies the subject of the sentence, TRNS-SADJ, the error is much higher. However, this is an artifact of a mistake in the Penglish grammar—the probability of the subject of the sentence being modified by an adjective was set to 1%, rather than 10%.

The CSCP model is also quite good at handling prepositional phrases. Prepositional phrases modifying the object of the sentence are considerably easier than those modifying the subject. This could be attributed to the fact that objects are more often modified by a PP than are subjects, by about a 4:1 ratio. However, this may not sufficiently account for the difference. An additional contributing factor may be that the model has trouble when an embedded phrase or clause intervenes between the subject and its verb. The model is also quite good at verb-modifying prepositional phrases, although slightly worse than it is at object-modifying PPs. Note that these two types of PPs are often syntactically ambiguous and must be resolved using semantic constraints. PP attachment ambiguities are an interesting topic, but investigations of them using the model will not be included in this thesis.

We turn now to the four-proposition sentences. A TRNS-ADJ is a transitive sentence in which both nouns are modified by adjectives. Similarly, a TRNS-NPP is one in which both nouns are modified by PPs. These sentences are considerably harder than the single-adjective or single-PP sentences, and are seemingly more so than should be expected given the addition of one more proposition. Presumably this is because, with two adjectives or PPs, the model has a tendency to confuse the attachments. This is supported by the fact that these sentence types have disproportionately high multiple-choice error. This confusion hypothesis can be confirmed by looking at sentences with one adjective and one noun-modifying PP. There are two variants of this. The adjective could modify the subject and the PP the object, or vice versa. In either case, the average strict error rates (13.8% and 13.1%) are considerably less than the error rates for TRNS-ADJ and TRNS-NPP (20.4% and 18.5%). The same is true using multiple-choice, where the error rates are 2.9% and 2.2% versus 8.8% and 4.0%. Therefore, we could hypothesize that the model experiences confusion when there are multiple attachments of the same type. There is less confusion when modifiers are of different types (adjectives versus PPs), even if both are modifying NPs.

So far we have considered only sentences with a single verb. The remaining sentence types all contain two clauses. The easiest of these is clearly the marked sentential complement, or SC. The CSCP model is surprisingly good at sentential complements. The strict error rate for sentential complement sentences is just 6.0%, better than that of many one-clause sentences. One reason for this is that sentential complements are quite frequent, occurring in 18% of all sentences, or about 212,000 times out of a million sentences. They may also be fairly easy because they are right branching and thus do not require very long-distance attachments. The case of reduced, and thus potentially ambiguous, sentential complements will be examined in more depth in Chapter 9.

The two CONJ- and the four SUB- sentence types involve coordinate and subordinate clauses with one intransitive clause and one transitive clause. They differ in which of these clauses comes first. The SUB- sentences also differ in whether the subordinate clause is first or second. The CONJ-TI sentences prove to be harder than the CONJ-IT. This can be attributed to a problem that occurs when any sentence ends in an intransitive—the end of the sentence is often a disambiguation point and the model does not have enough time to repair its mis-analysis. This same factor also explains the difference between SUB-R-TI and SUB-R-IT sentences. All four of these sentence types are somewhat difficult. This is probably because they are relatively rare and because they require an attachment to be formed that involves two verbs and a conjunction which are usually separated by other words. Nevertheless, the model still has less than 2.5% error according to the multiple-choice criterion on these sentences.

Even harder are the SUB-L-TI and SUB-L-IT cases, in which the subordinate clause precedes the main clause. In this case, the attachment of the two verbs using the conjunction is particularly difficult because there is a large separation between the conjunction and the second verb. The situation in which the intransitive clause precedes the transitive one is particularly bad because such sentences are ambiguous. The noun following the intransitive verb is not actually the object of the verb but the subject of the main clause. This is the NP/0 or subordinate clause ambiguity which will also be discussed further in Chapter 10.

The final two sentence types are the center-embedded relative clauses modifying the main subject. These are either subject-relative, RC-CS, or object-relative, RC-CO. Relative clauses, center-embedded ones in particular, can be difficult for a number of reasons. They have two transitive verbs whose subjects and objects can be easily confused. The center-embedded RC intervenes between the subject and its verb and creates a long-distance dependency. Also, most relative clauses are introduced by *that*, which is in itself a highly ambiguous word. *That* also serves as a demonstrative pronoun and to introduce sentential complements. All of Chapter 11 will be devoted to relative clauses.

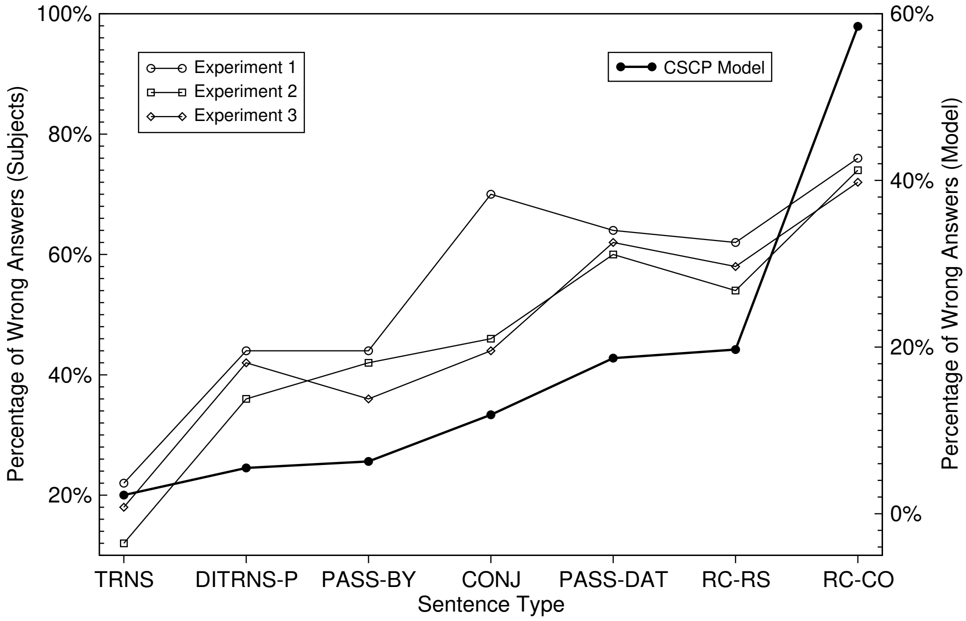


Figure 7.13: Comparison of the comprehension scores of aphasic patients from Caplan and Hildebrandt (1988) (Table 4.19) with those of the CSCP model.

7.3.1 Experiment 2b: Caplan & Hildebrandt (1988)

Caplan and Hildebrandt (1988) tested the comprehension ability of aphasic patients on a variety of different sentence types. In three separate experiments, subjects listened to sentences and demonstrated an understanding of them by acting out, using toys, the events described by the sentences. Caplan and Hildebrandt used nine different sentence types, six of which are among those used to evaluate the current model. These include TRNS, PASS-BY, DITRANS, CONJ, RC-RS, and RC-CO. An additional sentence type used in their study is a dative passive, PASS-DAT, which is a passive with both a by-phrase and a PP expressing a recipient. These do occur in Penglish, but the by-phrase always precedes the PP, which is opposite the normal ordering of such phrases in English. The other two sentence types used by Caplan and Hildebrandt cannot be tested on the current model because they involve cleft subjects and objects which do not occur in Penglish. The conjoined sentences used by Caplan and Hildebrandt involved a single subject with a compound verb phrase, as in, “The elephant hit the monkey and hugged the rabbit.” Because compound verb phrases also do not occur in Penglish, the complexity of these sentences was approximated by mixing CONJ-IT with CONJ-TI.

Figure 7.13 shows the pantomiming performance of the aphasic patients compared with the comprehension performance of the CSCP model. It is not clear whether it is best to model the task performed by the patients using the strict or multiple-choice criteria. The patients’ task is actually something like multiple-choice for the nouns, because the patients are given a limited set of dolls, but unconstrained for the verbs, which must be acted out. The strict criterion was used in assessing the network because the overall error rate of the patients was so high. The vertical axis on the left gives the error rate for the patients while the vertical axis on the right gives the error rate for the model. Because the tasks are quite different, and because the patients do have disordered syntactic comprehension, we should not expect an exact match between the patients and the model, but there does seem to be a general agreement between them. With the exception of the ordering of the PASS-DAT and RC-RS sentences, which are of similar difficulty, the model and the subjects agree in the rank ordering of the sentence types. The correlation between the model’s results on the seven data points and the average of the empirical results was 0.81. Aside from the fact that the overall error rate is much lower for the model, one major difference between the patterns of results is that the model has disproportionate trouble with the center-embedded object relatives. However, this could be due to a ceiling effect, as the patients may be very close to chance performance on the RC-CO sentences. The disproportionately high error rate for the model on these sentences may be due to their confusing propositional encoding, which is explained in Chapter 11.

7.3.2 Experiment 2c: Sensitivity to morphology

Word order is an important indicator of meaning, particularly in fixed word order languages like English. For example, it is often pointed out that "*John loves Mary*" does not mean the same thing as "*Mary loves John*." In a language that is not case marked, the comprehender must be sensitive to the order of the noun phrases to determine their thematic roles. Because the CSCP model is quite good at comprehending simple transitives, as well as ditransitives, it is clearly able to make use of ordering information to assign thematic roles.

However, free word order languages can signal thematic roles using inflectional case markings, rather than word order. Thus, we may ask if the model relies entirely on word order, or if it is also sensitive to inflectional morphology. We cannot test this in Penglish using case markings, because only pronouns are marked and scrambling of them is not permitted, but we can test sensitivity to the inflectional morphology of verbs. Sentences (46a) and (46b) are a simple pair that differ in deep structure but are distinguished at the surface level only in the verb's inflection. In order to determine that the boy is the agent in (46a) but the patient in (46b), the model must be sensitive not to word order but to the verb form.

(46a) The boy was killing yesterday.

(46b) The boy was killed yesterday.

To test the model's ability to comprehend such sentences, 151 sentence pairs were constructed, each sentence having a simple NP for the subject, either the past progressive or past passive verb form, and ending in the adverb *yesterday*. By necessity, all cases involved nouns that could serve as either the agent/experiencer of the intransitive form of the verb or the patient/theme of the passive form of the verb with no explicit agent.

The comprehension performance of the three models was tested on all sentences and their results averaged. Overall, the strict-criterion error rate was 10.8% for the intransitives and 10.9% for the passives, while the multiple-choice error rates were just 0.18% and 0.37%. One might expect the actives to be much easier than the passives, but their strict error rates were almost identical. This is attributable to the fact that the rare past progressive tense puts the actives at a partial disadvantage, making it more difficult to recall the tense of the action, which leads to errors in the strict criterion.

But the crucial comprehension question is whether the model can provide the relationship between the subject and the action, which is either agent/experiencer or patient/theme. Focusing on just this question, the strict error rate is 1.1% for the intransitives and 12.1% for the passives. When focusing on just this question, therefore, we do see an advantage for the active form over the passive form. But, crucially, the model is quite successful in both cases at deriving the proper thematic structure based purely on the verb inflection. It would be interesting to train and evaluate the model on a language with extensive scrambling and greater reliance on case marking.

As a side note, it is worth mentioning that the three networks differed in their preference for the active versus passive sentences. Adam and Chad preferred the passives, with 13.6% and 13.0% errors on the actives, but 9.3% and 9.5% errors on the passives. Bert, on the other hand, had only 5.7% errors on the actives but 13.8% errors on the passives. Therefore, there is the potential for different instances of this model to vary significantly in their performance on particular sentence types. The issue of individual differences is discussed further in Section 7.8.

7.4 Lexical ambiguity

In English, words can often play a variety of syntactic roles. It is quite common for the same surface form to act as either a noun or a verb, noun or adjective, or all three. Sampling in the range from the 200th to the 299th most common words that appear in newsgroup discussions, I find 20 words that can only be used as nouns, 7 unambiguous verbs, and 14 unambiguous adjectives. But there are also 35 words that can be either nouns or verbs, 5 noun/adjectives, one verb/adjective, and one word, *mean*, that can be a noun, verb, or adjective. Intuitively, such ambiguities, frequent as they are, seem to cause relatively little difficulty to human comprehenders. So we might ask, how well can the model handle lexical ambiguity?

Noun	TRNS Subject	TRNS Object
answer	288	2,030
bird	344	230
bat	350	255
fish	463	339
cat	499	271
question	693	2,297
dog	1,599	808
test	–	897

Table 7.2: Frequency of some nouns as the subject or object of TRNS sentences.

7.4.1 Experiment 3a: Noun/verb lexical ambiguity

Penglish contains three words—*fish*, *park*, and *question*—that can be used as either nouns or verbs. One way to test whether these words cause confusion for the model might be to compare the average comprehension performance on sentences using these words with that on all sentences or on sentences not using these words. However, there are several potential confounding factors in this approach. The lexically ambiguous words may be more or less frequent than other words and thus easier or harder to process for purely statistical reasons. The two samples may also contain systematically different distributions of sentence types. For example, *fish* and *park* are commonly used as intransitives, so the model may have difficulty with sentences using them because it is intransitively-challenged. Even if we control for overall length, we might have one set of sentences with many prepositional phrases, which are quite easy, and another set with many relative clauses, which are hard. This scenario is particularly likely in the case of *park* because the noun sense often appears in a PP expressing location or destination.

Syntactic complexity can be controlled only by comparing sentences with identical syntactic structure. However, we still need to worry about frequency because we may be using verbs in unusual syntactic structures. In an ideal world, we could compare the comprehensibility of *fish*, *park*, and *question* to that of unambiguous words that are matched both for overall frequency and for frequency in the tested constructions. However, Penglish does not have quite enough words in its lexicon for that. Therefore, the ambiguous words will be compared to a selection of other nouns and verbs that are close to them in frequency. Because the frequency variable is not completely accounted for, we still must consider its possible effect in interpreting the results.

If the model is sensitive to lexical ambiguity, that sensitivity should have its effect on comprehension but not on semantic encoding. The semantic encoder is operating on word meanings and knows nothing about a word's surface form. It has no access to the fact that the verb *park* and the noun *park* happen to be pronounced the same, and any lexical differences that appear during semantic encoding and decoding cannot be attributed to surface-form ambiguity. Therefore, in looking for ambiguity effects, we would like to distinguish errors caused by the CPP system during sentence processing, from those caused by the semantic system by its failure to adequately encode or decode sentence meaning.

There are two ways we might separate comprehension errors from encoding errors. One is to subtract the question-answering error rate that results from encoding and decoding the sentence meaning from the error rate that results from comprehending the sentence and then decoding the induced meaning. This difference will be called the *comprehension-induced error* (CIE). Of course, this measure is not perfect. It is possible that a mistake in comprehension produces no effect on question answering because the decoder can cover for it. Conversely, if the decoder were very sensitive, a small comprehension error might result in a big increase in error rate. But, on average, the measure is still probably quite meaningful.

A second measure of comprehension-induced problems is simply to look at the difference between the message that resulted from encoding the sentence and the message that resulted from comprehending the sentence. The sum-squared difference between these two representations should reflect the degree to which the comprehension system has failed to do its job. This measure will be called the *comprehended message error* (CME).

And now for some results. We begin by looking at the noun forms of the ambiguous words. *Fish* and *question*, along with several unambiguous nouns, were tested in two different contexts, as the subject or as the direct object in

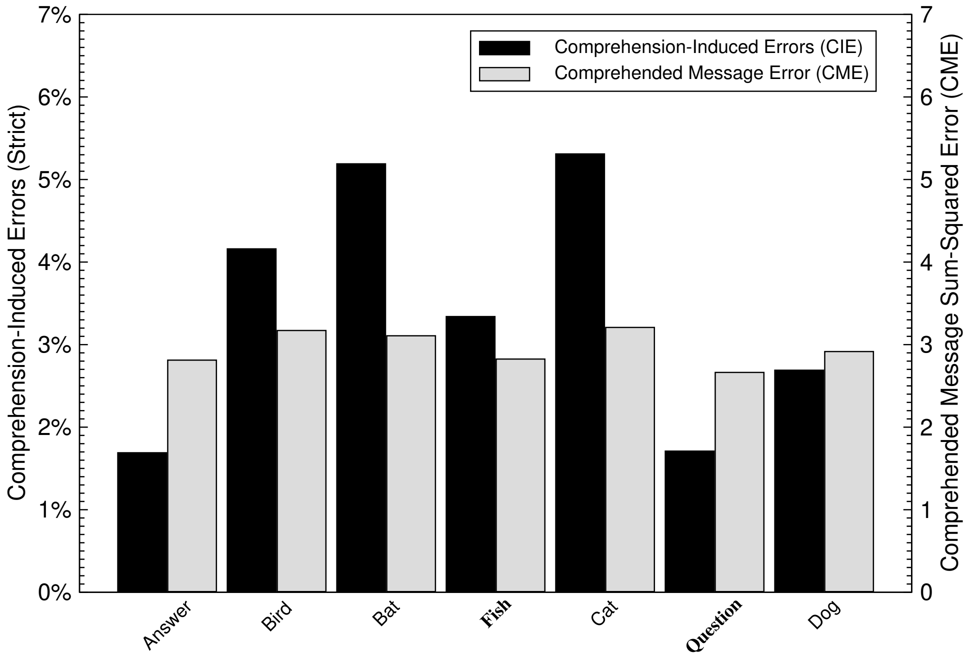


Figure 7.14: Two measures of comprehension error for simple transitives with various nouns as subjects.

simple transitive sentences (TRNS). Table 7.2 shows the frequency, out of 1 million Penglish sentences, with which a simple transitive sentence occurs with each of the nouns as its subject or direct object. *Fish* is of moderate frequency. *Question* is fairly frequent as a subject and quite frequent as an object, where it is similar in frequency to *answer*.

Figure 7.14 shows the CIE and CME error measures for the TRNS sentences using each of seven nouns as subjects. For both measures, lower is better, indicating less error introduced during comprehension. The results are averaged over all three networks, processing 200 sentences in each condition. The nouns are ordered from lowest to highest frequency. *Fish* turns out to be easier than any of the neighboring animals, *bird*, *bat*, and *cat*. *Question* is also handled quite well, resulting in less comprehension error than the more frequent *dog*, and very similar error to the semantically similar word *answer*.

Figure 7.15 shows similar measures for the nouns being used as direct objects, rather than subjects. In this case, there seems to be a greater sensitivity to frequency, with the funny exception of *bird*. *Fish* is processed more easily than the less frequent objects, but with more difficulty than the more common objects. *Question* results in the least comprehension error, although it also happens to be the most frequent word. In summary, based on the evidence in these two figures, there does not appear to be any tendency for the model to experience comprehension difficulty resulting from a noun that can also play the role of a verb.

Park was not included in the previous tests because it rarely acts as a subject or object. Whether or not it is true of English, Penglish has the property that places like *park* tend to appear most often in prepositional phrases. Therefore, comprehension of the places *park*, *zoo*, *school*, and *house* was tested in the context of transitive sentences with a verb-modifying prepositional phrase, TRNS-VPP, expressing either location or destination. The noun of interest always served as the object of the preposition. The relevant frequencies are shown in Table 7.3 and the results are shown in Figure 7.16. The results of this experiment are somewhat unclear as there is a wide range of frequencies between the nouns. *Park* is fairly easy as measured by CIE. Its CME error is about the same as that for *zoo*, although *zoo* is somewhat less common. However, the CME is not that much lower for *school* and *house* even though they are 5 and 8.5 times more common. Again, there does not seem to be a significant disadvantage for lexical ambiguity in comprehending nouns.

We have seen that the model is fairly good at comprehending these ambiguous words when they are used as nouns. But it happens to be the case that all three of them are much more common as nouns than they are as verbs. It is possible that the model is just assuming the words are always nouns, and sacrificing its ability to handle them as

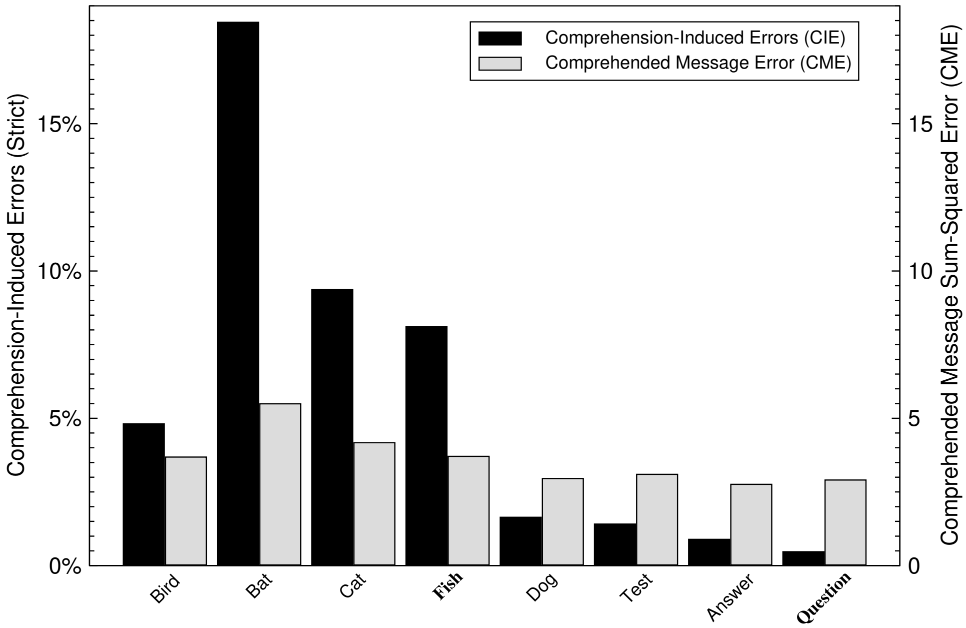


Figure 7.15: Two measures of comprehension error for simple transitives with various nouns as direct objects.

Noun	TRNS-VPP Object
zoo	144
park	687
school	3490
house	5885

Table 7.3: Frequency of some place-nouns as the objects of PPs in TRNS-VPP sentences.

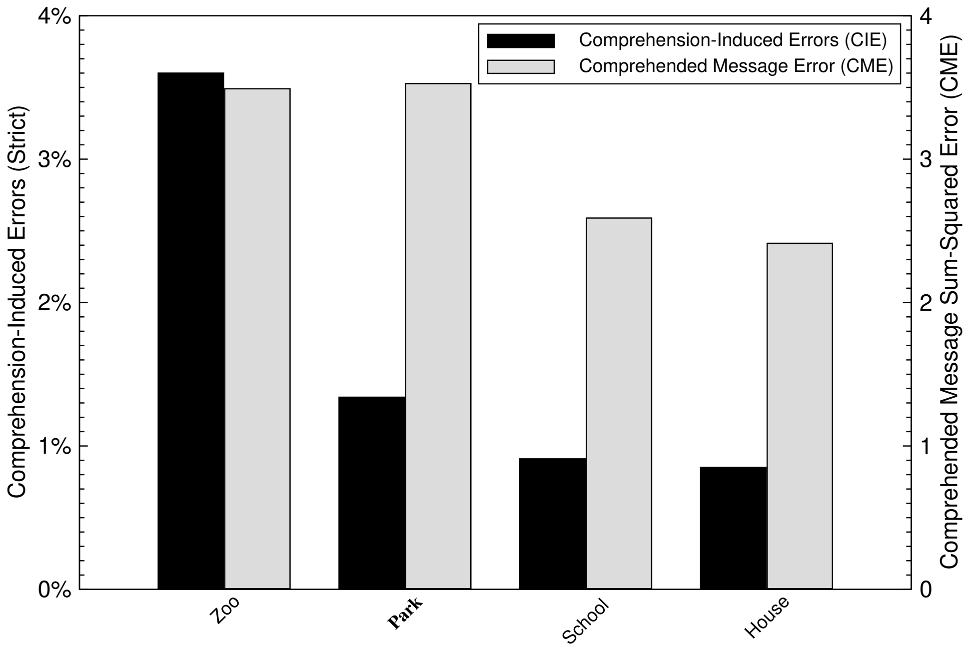


Figure 7.16: Two measures of comprehension error for TRNS-VPP sentences with various nouns as the object of a location or destination PP.

Noun	INTR-VPP	TRNS
bite	1.1	73
forget	2.1	35
park	2.9	5.0
guess	5.8	31
bark	6.6	–
throw	10.3	143
fish	16.2	–
hope	59	–
drive	104	152
fly	142	2.6
question	–	126

Table 7.4: Frequency of some verbs in PP-modified intransitives and in transitives.

verbs. Therefore, we will now look at verb usage. Two sentence types are used here, simple transitives (TRNS) and intransitives with verb-modifying PPs (INTR-VPP). *Park* can be used as either, but *fish* must be intransitive and *question* must be transitive. Table 7.4 gives the frequency of these and some other verbs in the two sentence types.

As shown in Figure 7.17, *park* has very low CIE and quite low CME given its low frequency. *Fish* does result in higher comprehension error than the somewhat less frequent *throw*, but it is not nearly as bad as *guess* and *bark*. The ambiguous verbs do not stand out as particularly difficult intransitives. Figure 7.18 shows the results for the transitive verb uses. *Park* causes more comprehension errors as a transitive than as an intransitive, but it is only slightly worse than the much more frequent *guess* and *forget*. The CIE for *question* is in line with its frequency, but the CME is somewhat elevated. Although the results for verb usage are not quite as clear as those for noun usage, overall there seems to be little indication that the CSCP model is significantly affected in its comprehension performance by noun/verb lexical ambiguity.

7.4.2 Experiment 3b: Adjective sense ambiguity

The previous experiment involved lexical ambiguity between words of different syntactic class. Another type of lexical ambiguity is that between multiple senses of a word that share the same syntactic class. Penglish contains a number of verbs that have multiple senses. However, it is difficult to test the model's ability to distinguish the verb senses because most of them differ not only in meaning but syntactically, in the argument structures that they project. It would be difficult to adequately control for the effects of argument structure frequencies in looking for effects of sense ambiguity. However, one form of lexical ambiguity that we can test is that of the two adjectives, *hard* and *nice*, whose meaning depends on the noun being modified. In Penglish,*hard* can either mean difficult (*hard_D*) or not soft to the touch (*hard_T*), while *nice* can mean kind (*nice_K*) or unspoiled and new (*nice_N*).

Testing the comprehensibility of adjectives is relatively easy, but we still need to be aware of possible frequency effects. Table 7.5 lists the 12 adjectives used in Penglish, along with their frequencies per 1 million sentences. The ambiguous adjectives are at the low end of the frequency scale. The comprehensibility of the adjectives was tested in the context of randomly sampled 3-prop sentences. The majority of these sentences are transitives with the adjective modifying the direct object, although in some cases the adjective modifies the subject or the object of a PP if the verb is intransitive.

Figure 7.19 shows the comprehension-induced error (CIE) and the comprehended message sum-squared error (CME) for sentences using the various adjectives. With the exception of *fierce*, differences between the adjectives are relatively small. The ambiguous adjectives *hard_D*, *hard_T*, and *nice_N* all result in lower comprehension-induced error than many of the more frequent adjectives. *Nice_K* does result in slightly higher CIE than either of its neighbors, *young* and *mean*, but not significantly so,[7] and the differences in CME are certainly not significant. Therefore, the CSCP model does not have any detectable problem comprehending adjectives with multiple senses even though such

[7] *nice_K* vs. *young*: F(1,10726)=1.538, p = 0.215; *nice_K* vs. *mean*: F(1,10693)=3.258, p = 0.071

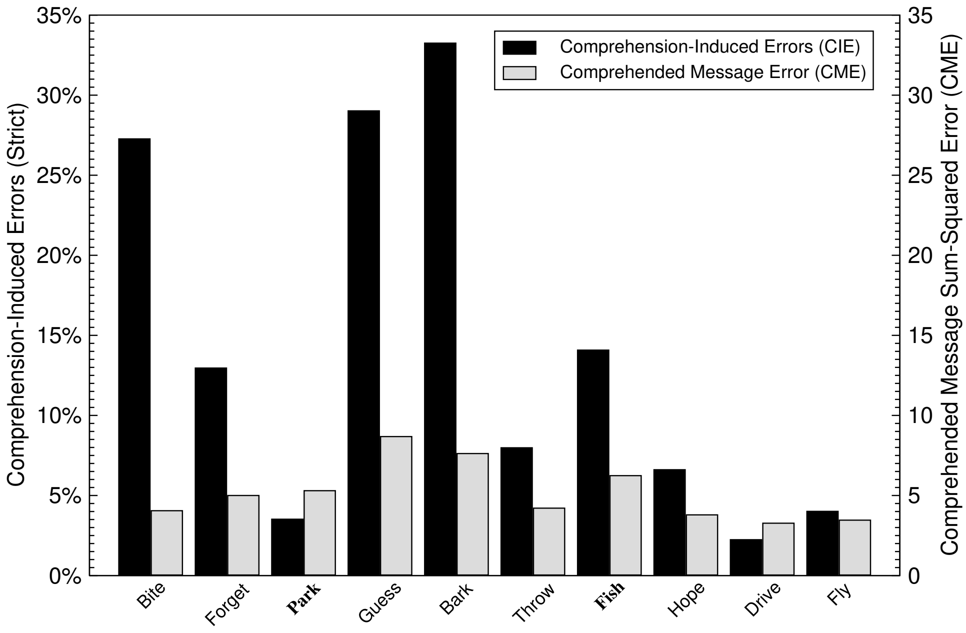


Figure 7.17: Two measures of comprehension error for various verbs in PP-modified intransitives.

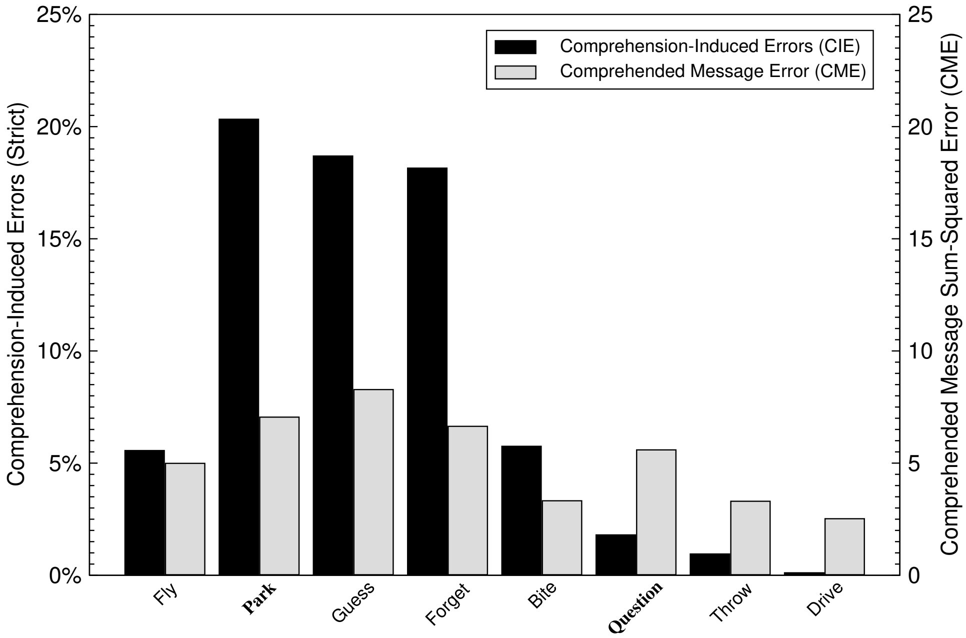


Figure 7.18: Two measures of comprehension error for various verbs in simple transitive sentences.

Adjective	Frequency
fierce	7,854
hard_D	13,781
hard_T	14,044
nice_N	17,097
young	21,049
nice_K	21,778
mean	22,339
loud	25,331
small	37,146
big	44,358
new	48,900
old	52,299

Table 7.5: Frequency of the adjectives per 1 million Penglish sentences.

adjectives are temporarily ambiguous and the correct meaning is dependent on the following noun.

7.5 Experiment 4: Adverbial attachment

This next test of comprehension focuses on a difficult adverbial attachment ambiguity. Consider the following sentences:

(47a) I said that I will come tomorrow.

(47b) I will say that I came yesterday.

(48a) I said that I will come yesterday.

(48b) I will say that I came tomorrow.

Each sentence contains two verbs, and the adverb could potentially attach to either one. What determines the appropriate attachment is the tenses of the verbs, in conjunction with our understanding of the meaning of the words *yesterday* and *tomorrow*. Sentences (47a) and (47b) are cases of *low attachment*. That is, the adverb correctly attaches to the most recent verb, which is lower down in the (inverted) parse tree. These sentences are intuitively not too difficult, although (47b) seems a bit harder, possibly because it forces the reader to create a scenario in which the writer might be planning to tell a lie. In contrast, sentences (48a) and (48b) seem much more difficult. Most readers will garden-path on the first reading, but should be able to figure out the correct meaning on reanalysis. Sentence (48b) is actually globally ambiguous. It is possible to get a low-attachment reading under the assumption that the *saying* will be farther in the future than the *coming*, and this may be the preferred reading for some people. The possibility of such multiple readings is something to consider if experiments are to be conducted on such sentences using English. However, Penglish holds to the rule that *yesterday* only modifies past tense verbs while *tomorrow* only modifies present or future tense verbs, so we need not worry about that for now.

In order to test the ability of the current model to resolve such ambiguities, forty sentences were constructed in each of the four conditions. All sentences involved a transitive sentential complement and ended with either *yesterday* or *tomorrow*. One of the verbs in each sentence was in a past tense form and the other in a future form. A set of 40 base sentences satisfying these conditions was drawn at random from a Penglish corpus, and the items for the four conditions were created by modifying the verb tenses and adverbs appropriately. Of the 40 sentential complements, 27 were marked and the others were reduced. Sentences (49a)–(49c) are the first three low-attachment sentences using *tomorrow* and (50a)–(50c) are the corresponding high-attachment versions. On the surface, low-attachment sentences using *tomorrow* were identical to high-attachment sentences using *yesterday* with the exception of the adverb.

(49a) The manager knew I will read the answers tomorrow.

(49b) The boy believed the cop will ask us tomorrow.

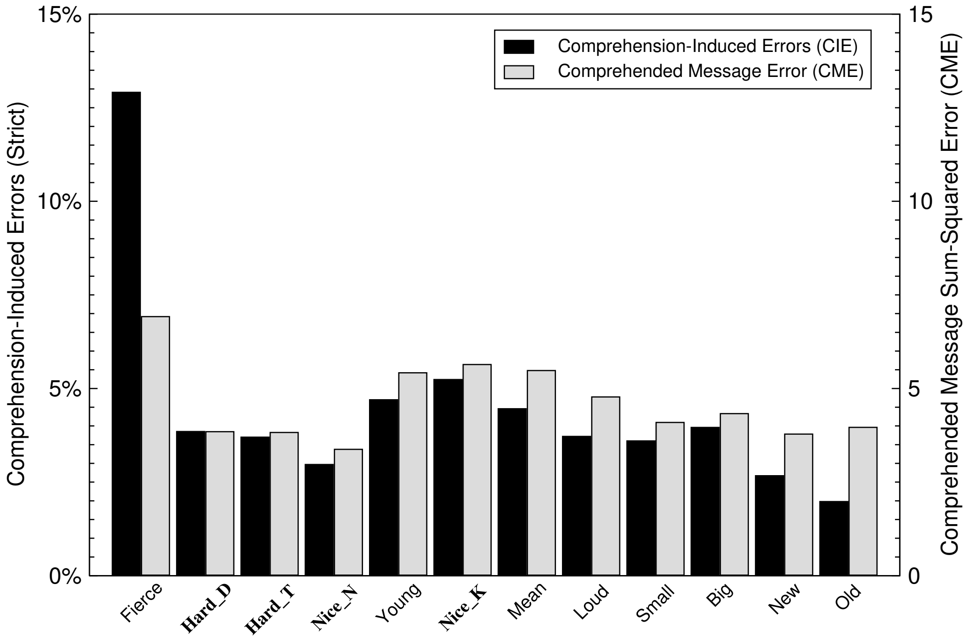


Figure 7.19: Two measures of comprehension error for various adjectives in 3-prop sentences.

(49c) Teachers hoped boys will take you tomorrow.

(50a) The manager will know I read the answers tomorrow.

(50b) The boy will believe the cop asked us tomorrow.

(50c) Teachers will hope boys took you tomorrow.

The overall frequency of transitive sentential complements that end in either *yesterday* or *tomorrow* and that have either *high* or *low* attachment actually varies quite a bit. With the adverb *yesterday*, the frequency of such a sentence with low attachment is 610 (per million) while that with high attachment is 498. The frequencies of low and high attachment using *tomorrow* are 262 and 24, respectively. The reason high attachment is so rare for *tomorrow* is that verbs that take sentential complements, which are often verbs of thinking or communicating, are rarely used in the future tense.

Figure 7.20 gives the message encoding and comprehension error rates for the four sentence types using the strict criterion. Note that the error rate is not all that high, given that these are 5-prop sentences. As might be expected, the comprehension error rate for the high-attachment sentences is higher than that of the low-attachment sentences. The error for the less frequent *tomorrow* is also greater than that for *yesterday*. However, somewhat strangely, the difference in error rates resulting from the attachment site is even stronger for the encoder than it is for comprehension. The encoder should not be sensitive to the syntactic nature of the attachment. However, the reason for the difference in encoding error may be that verbs that take sentential complements receive adverbial modification less often than the verbs in the low attachment position. Apparently this was not a well-enough controlled experiment.

The results in Figure 7.20 may seem to imply that the model is only slightly worse at high attachments than it is at low attachments. However, that is not the case. Those results are averaged over all 15 possible questions that we might ask about each sentence's meaning. What happens if we focus on the difficult question about the site of attachment: "Which verb is modified by *yesterday*?" Figure 7.21 gives the comprehension and encoding error rates on this question alone. Although the model gets the low attachment answer correct 68% of the time, it almost never gets the right answer in the high attachment case. Of course, this cannot be entirely attributed to syntactic processing on the basis of this experiment because of the confound with semantic plausibility. The error rate for the encoder is also quite high in the high-attachment condition. Incidentally, in a high-attachment situation, even if the model could not tell you the verb that the adverb is modifying, if asked which adverb is modifying the high verb, it can still provide the correct answer 84% of the time (strict criterion). Thus, if prompted, the model is willing to attach the adverb to either

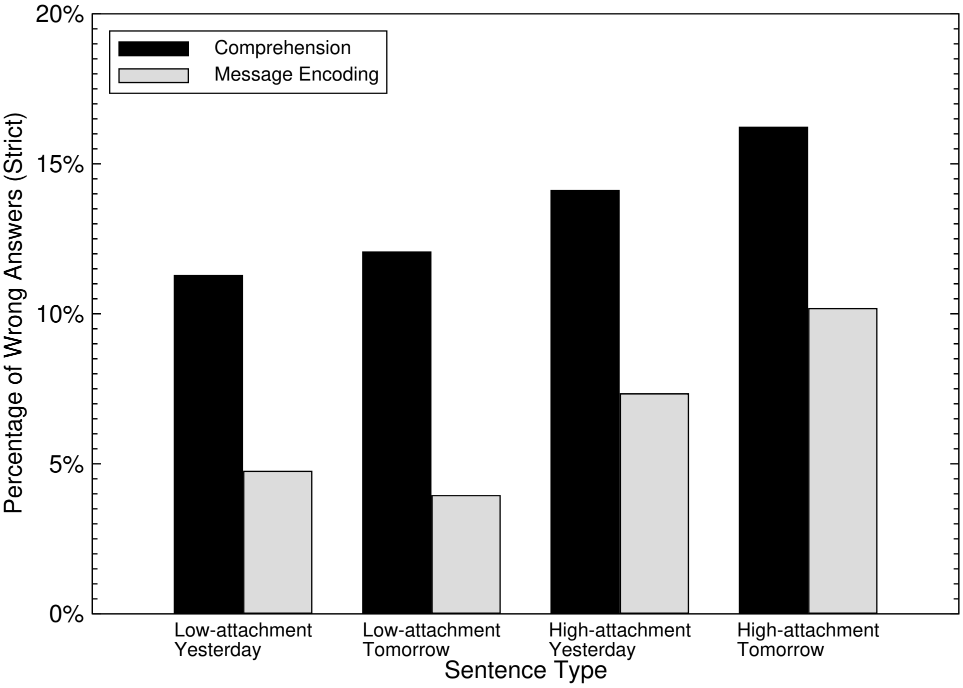


Figure 7.20: Overall comprehension and encoding performance on the ambiguous adverb-attachment sentences.

verb.

A researcher looking at results such as those in Figure 7.21 might be tempted to conclude that this system possesses a rule against high attachment of adverbs. Rules of the human sentence processing mechanism have been proposed on the basis of far weaker evidence. However, although it seems to possess rule-like behavior in this case, we know that no such rule has been built into the architecture of the CSCP model. Its severe impairment in comprehending high attachments arises from its sensitivity to a combination of factors that all support low-attachment readings. Principle among these are frequency biases. We have already seen that high attachments are less common in sentential-clause sentences ending in an adverb. However, the frequency difference between high and low attachments in this situation is not nearly as strong as it is in the Penglish language as a whole. Of all adverbs that occur in a 1 million word corpus of Penglish, 13,447 receive high attachments, meaning that another verb intervenes between the adverb and the verb it modifies. But 116,405 of the adverbs have low or local attachments. Thus, low attachments are, overall, nearly 9 times more common.

Another main factor working in favor of low attachments is locality. It seems likely that the model will have trouble associating an adverb with its verb if other material intervenes. To really test the effect of locality in this model, one would have to create a language in which all other factors, including frequency, are balanced. Finally, in the case of the current experiment, two other factors affecting the outcome were the degree to which sentential complement verbs are likely to appear in future tense and the degree to which they are modified by temporal adverbs. It remains to be seen if humans would have quite the same level of difficulty in their first-pass reading of high-attachment sentences similar to those used here.

Incidentally, I cannot resist mentioning that finding high-attachment adverbs is difficult if not impossible with TGREP, but can be done with a fairly simple pattern using TGREP2:

```
ADVB=a > (* < (/^VERB/.. (/^VERB/ .. =a)))
```

This pattern looks for an adverb (`ADVB`), which it labels `a`. The parent verb phrase of this adverb, matched by `*`, must produce a verb (the one modified by the adverb), which is followed by another verb that is itself followed by the original adverb (accessible through the label `a`). That is, another verb must intervene between the adverb and the verb it modifies. Replacing the final `=a` with `ADVB` would not work because that might match a different adverb later in the sentence. The reason this pattern cannot be expressed in TGREP is that the modification and intervening relationships

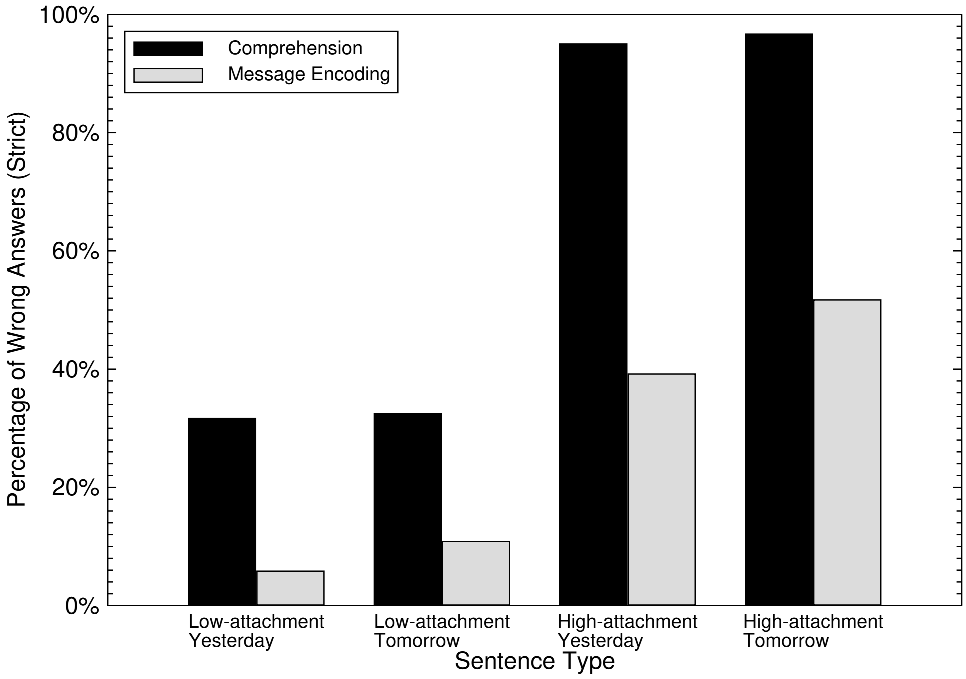


Figure 7.21: Comprehension and encoding performance in answering the question, "Which verb is modified by the adverb?"

are circular and TGREP only permits relationships with a tree structure. Appendix C discusses TGREP2 in greater detail.

7.6 Experiment 5: Prepositional phrase attachment

As discussed in Section 3.6, prepositional phrase attachment ambiguities have been an important area of study in psycholinguistics. One of the most common forms of attachment ambiguity occurs when a PP follows a direct object as in (51a) and (51b), and could potentially modify either the main verb or the direct object. Simple theories of attachment preferences, such as an initial bias toward verb attachment (Rayner et al., 1983), do not seem to explain all of the available data. More recent theories depend on the status of the PP as an argument or modifier, with arguments receiving preferred attachment (Schütze & Gibson, 1999).

(51a) The spy saw the cop with binoculars.

(51b) The spy saw the cop with a revolver.

Originally, it was hoped that the Penglish language would be sufficient to allow us to address questions of argument versus modifier status. However, on further inspection it seems that there are simply not enough lexical items to make this practical. Because the model is expected to be quite sensitive to frequency at a variety of levels, the language on which it is trained must be representative of English in this respect. The levels that may be important include, among others, the overall probability of a particular preposition modifying a verb or noun, the overall probability of a particular preposition playing a certain thematic role (possession vs. instrument for *with*), the probability of individual nouns and verbs being modified by various PP types, the probability of individual nouns serving as the object of the preposition, as a function of the modified word and thematic role, and so on. The grammar of Penglish constrains, as an approximation to English, the modification probability of each noun or verb by each type of PP. However, the number of lexical items is limited. Therefore, Penglish is unlikely to be representative of English in more general measures, such as the overall frequency of occurrence of each preposition. A much larger vocabulary is necessary if a detailed study of PP attachment is to be done using the current model.

However, we can in the meantime ask some more basic questions. For example, to what extent is the model sensitive to potential PP attachment ambiguities? Does it have a strong VP or NP attachment bias? Is it able to form proper attachments on the basis of the preposition used? Is it able to resolve attachments that are only disambiguated by the object of the preposition, as in (51a) and (51b)?

In order to answer these questions, six sentence types were constructed, three with correct NP attachments and three with correct VP attachments. All sentences used a transitive verb with a PP following the direct object. The first pair of sentence types are called *unambiguous* because they are disambiguated at the preposition. That is, they used prepositions that, in Penglish, can only modify either NPs or VPs, but not both. The exclusively VP-modifying prepositions are *to* and *in*, while the exclusively NP-modifying prepositions are *by* and *of*. Strictly speaking, *of* can be used to modify certain verbs, as in "*I've heard of you*," but in Penglish it cannot follow the direct object in modifying a transitive verb, and is thus unambiguous in this context.

The second pair of sentence types used the prepositions *for*, *with*, and *on* and were thus ambiguous at the preposition, but were resolved unambiguously by the following NP, as in (51b). These are called *locally ambiguous*. The final pair of sentence types used the same prepositions but were *globally ambiguous*, in that the object of the preposition did not fully resolve the attachment, although it may have lent a bias toward one reading. In (51a), for example, the binoculars are most likely the instrument, but they could be a possession of the cop as well.

All sentence types were drawn from a large corpus of parsed Penglish. In the case of the globally ambiguous sentences, the "correct" attachment site was determined by the form of attachment in the corpus sentence. Sentences using the ambiguous prepositions were re-parsed using the SLG program (Appendix B) to determine if they were locally or globally ambiguous. Two hundred sentences were used in the unambiguous and locally ambiguous classes, but for the globally ambiguous sentences there were 125 NP-modifying and 185 VP-modifying. All three networks were tested and their results averaged. In testing the model's ability to resolve the ambiguity, the critical comprehension question is, "What does the object of the preposition modify?," the correct answer being either the action or direct object.

It might be reasonable to use a multiple-choice error measure to evaluate the model's performance. However, the model was set up to use only multiple-choice foils of the same concept class as the correct answer. In this case, we would like to offer the object and action as foils for one another, as they are the two most likely alternatives, but they are of different classes. Rather than alter the multiple-choice criterion, the strict criterion was used.

Figure 7.22 shows the strict-criterion error rates in answering the critical attachment question for each of the six sentence types. First of all, note that the error rates in the unambiguous conditions are very low, at 4% for the NP-modifying and 0.5% for the VP-modifying. The model has little trouble with either unambiguous attachment, although it seems to have a VP-modification preference. For the locally ambiguous sentences, the error rates are worse, as might be expected, but the model still does quite well. Errors are even more common on the globally ambiguous sentences, but the model is still getting the correct attachment on about 74% of the trials, where chance performance is virtually 0% correct in the strict criterion. Many of these errors may not reflect wrong attachment sites as much as they do minor errors in producing the action or object representation. A multiple-choice measure with appropriate foils would undoubtedly result in considerably lower error rates.

The fact that the networks perform quite well even on the globally ambiguous sentences indicates that they are able to use weak semantic and/or statistical biases to their advantage in resolving such ambiguities. The fact that the locally ambiguous sentences are easier confirms that the model is able to use semantic biases to resolve structural ambiguities. Finally, the very low error rate on the unambiguous sentences indicates that the model has internalized the modification rules implicit in the Penglish language.

7.7 Reading time

This section provides an introduction to the properties of the simulated reading time (SRT) measure used in the CSCP model. This should be helpful in interpreting the reading times as they appear in later chapters. As explained in Section 6.5.4, the SRT consists of a weighted average of four components: the word stem prediction error, the word ending prediction error, the message change, and the message activation. In brief, the word stem and ending prediction errors reflect the divergence error of the two parts of the previous word prediction. The message change is the root mean squared difference between the previous message representation and the new one. The message activation is the

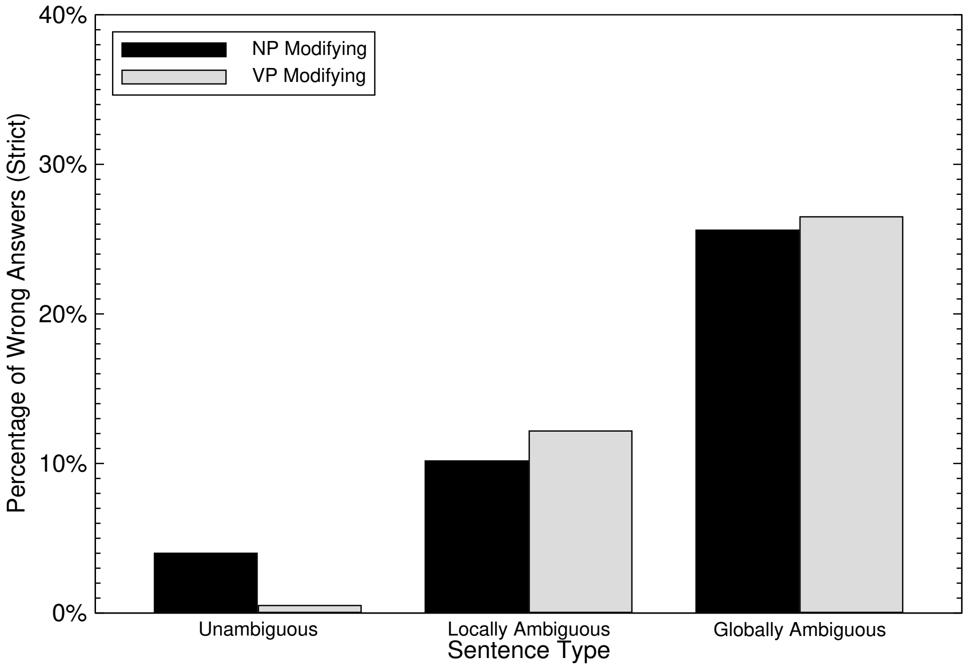


Figure 7.22: Error rates in answering the attachment-site question when comprehending the six PP sentence types.

average activation level of the units in the message representation. The components are normalized to have average values of 1.0 and then averaged, using the proportions shown in Table 6.2 (22:3:50:25) to established the SRT.

Figures 7.23–7.27 plot the SRT and its components during the comprehension of several different sentence types. The SRT, along with its four components, is plotted for each word position in the sentence. The units differ among the various components, but all of them have been normalized to mean values of 1.0. The reading times displayed in the figures have been averaged across many sentences that all share the same syntactic structure. The sentences listed at the bottom of the figures are just representative of the sentence class.

Figure 7.23 shows the reading times for simple transitive and intransitive sentences. The word-stem prediction and the word-ending prediction are both quite low on the articles, somewhat higher on the nouns, and highest on the verbs. The word-ending prediction is subject to violent spikes when it encounters unexpected verbs. However, it accounts for only 3% of the SRT, so these have a relatively small effect on the overall reading time. Both prediction measures are quite low on the period—the model was not surprised by the ends of the sentences—although the end of the intransitives was more surprising than that of the transitives.

The message change is fairly high on the content words, with verbs causing more change than nouns. Messages sometimes also change quite a bit at the end of the sentences, particularly for the intransitives. This provides direct support for the hypothesis that the model needs to restructure its representations when it is surprised by the end of an intransitive. Interestingly, over the course of these simple sentences, the average message activation actually drops. That is particularly true of the intransitives. This may be an indication that the model was initially prepared to encounter a more complex message, and began to represent multiple potential interpretations in parallel. As the sentences played out in their simplest possible forms, the messages could actually be simplified.

The reading times for sentences with right-branching subject-relative RCs and center-embedded object-relative RCs are shown in Figures 7.24 and 7.25, respectively. The prediction measures are again highest on the verbs and next highest on the nouns. In the right-branching case, the second verb is somewhat harder than the first. In the center-embedded case, the prediction of the main verb is very difficult. Interestingly, the message change measure reaches its peak when the word *that* occurs. This suggests that the model is preparing in advance for the upcoming relative clause when it encounters a *that*. In this case, the average message activation grows over the course of the sentences. For these relative clause sentences, the overall SRT is highest on the verbs and on the relative pronouns.

In the case of the sentential complement sentence in Figure 7.26, the message change is again quite high on the

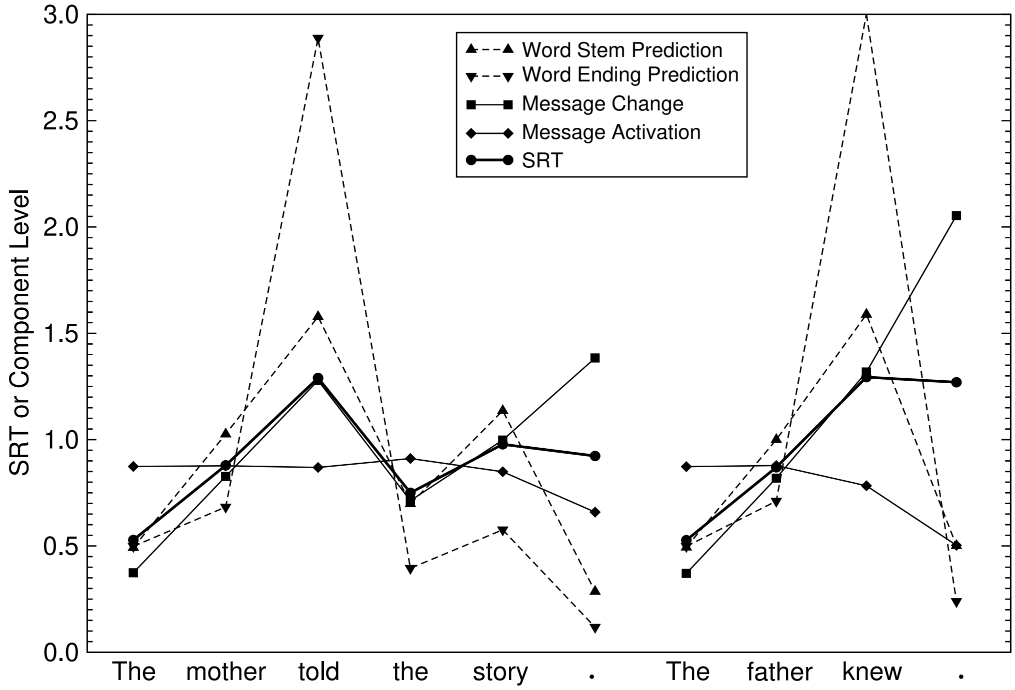


Figure 7.23: The SRT reading time measure and its components during simple transitive and intransitive sentences.

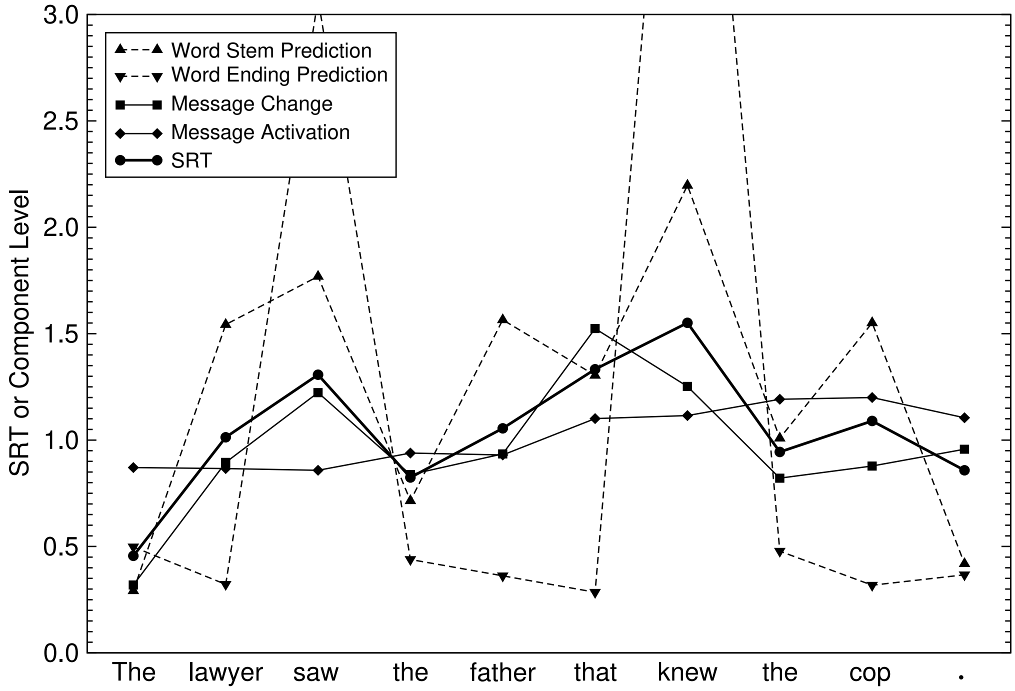


Figure 7.24: The SRT reading time measure and its components during right-branching subject-relatives.

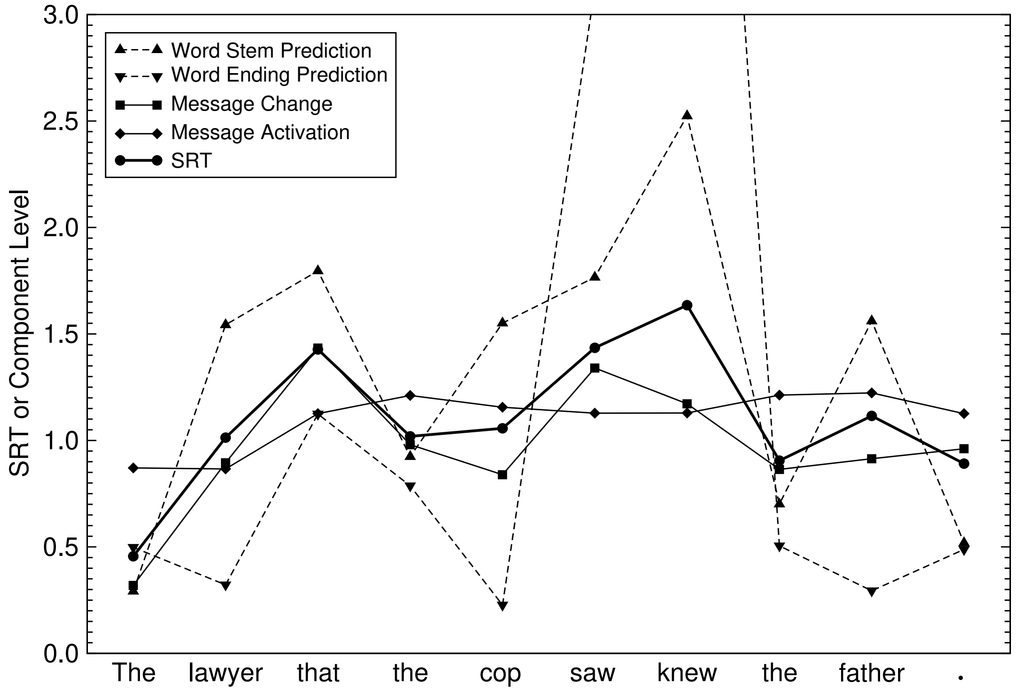


Figure 7.25: The SRT reading time measure and its components during center-embedded object-relatives.

word *that*, indicating that it has begun to prepare for the sentential complement clause in advance. Interestingly, all of the reading time measures, with the exception of the message activation, have almost identical values while reading the subject and verb in the SC as they do when reading the main subject and verb. This suggests that the model recognizes the fact that sentential complements have almost identical structure to main clauses. From a syntactic standpoint, when the SC begins, it is as if the grammar has returned to its initial state, after which any sentence could occur.

Finally, Figure 7.27 shows the reading time measures during the course of a sentence with a right-hand subordinate clause. Note that the middle prediction spike is due to the word-stem prediction, while the outer spikes are due to the word-ending prediction, and are thus of less relevance. The model makes a long pause on the subordinating conjunction. Not only is this due to its being unexpected, there is also a large change in message when the conjunction is encountered. The model must perform a major restructuring of the message to handle the additional clause. Other than that, reading time is predicted to be similar during the subordinate clause and during a main clause, although in the latter it will be overall somewhat slower due to the increased density of the message representation.

The SRT measure used here is only a rough first attempt at extracting reading times from the CSCP model. As discussed in Section 6.5.3, it was based on a theory of factors that are believed to affect sentence comprehensibility and thus reading times. However, the four components of the SRT do not adequately represent those factors and little effort was made to fine-tune their weightings to match actual human reading times. Although a linear combination of the components seemed to be the most reasonable first try, introducing non-linear transformations of the components, or interactions between them, may allow the SRT to more closely model the settling times that would occur if the CSCP were a truly dynamical, fully-recurrent network. Nevertheless, it is hoped that the current version of the SRT will be sufficient to look at relative differences in reading times between specific conditions in well-controlled experiments. Producing an accurate match to human reading times will be an interesting project for the future.

7.8 Individual differences

The three versions of the CSCP model, Adam, Bert and Chad, all share the same architecture and were trained on the same sentences using the same methods. The only difference between them was the assignment of random initial link weights. The main reason for training three networks was to look for individual differences between them and to try

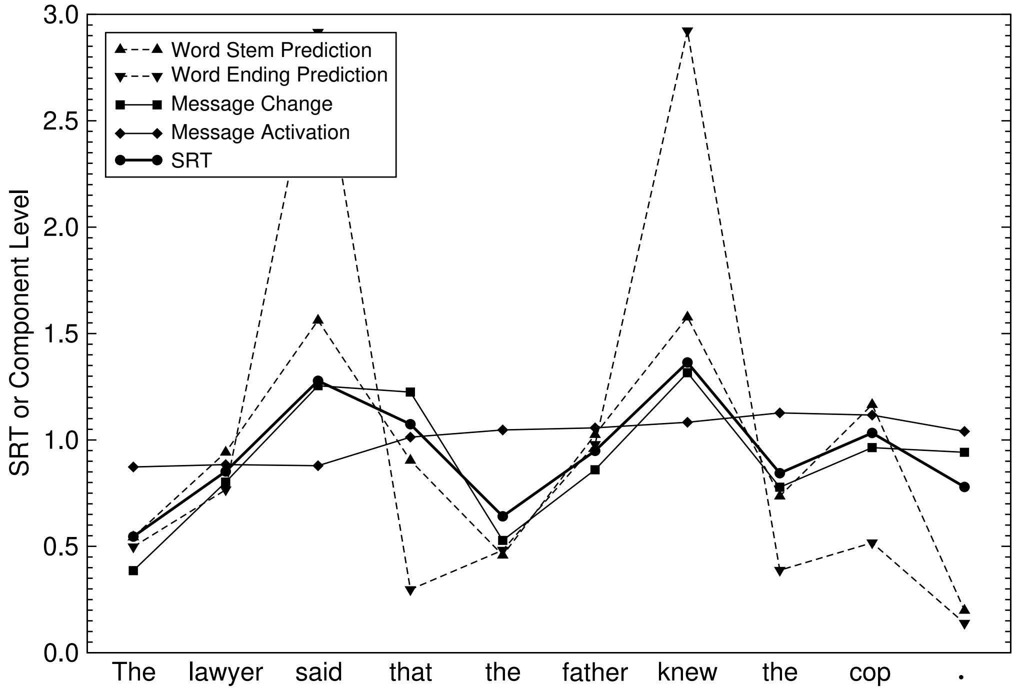


Figure 7.26: The SRT reading time measure and its components during sentential complements.

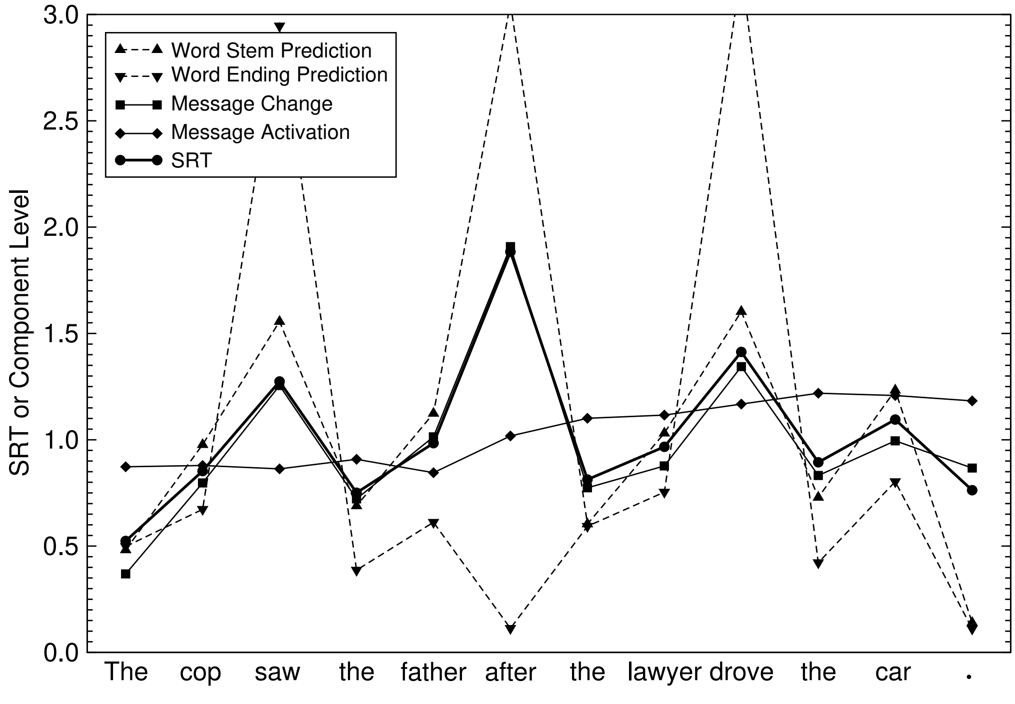


Figure 7.27: The SRT reading time measure and its components during a right-hand subordinate clause.

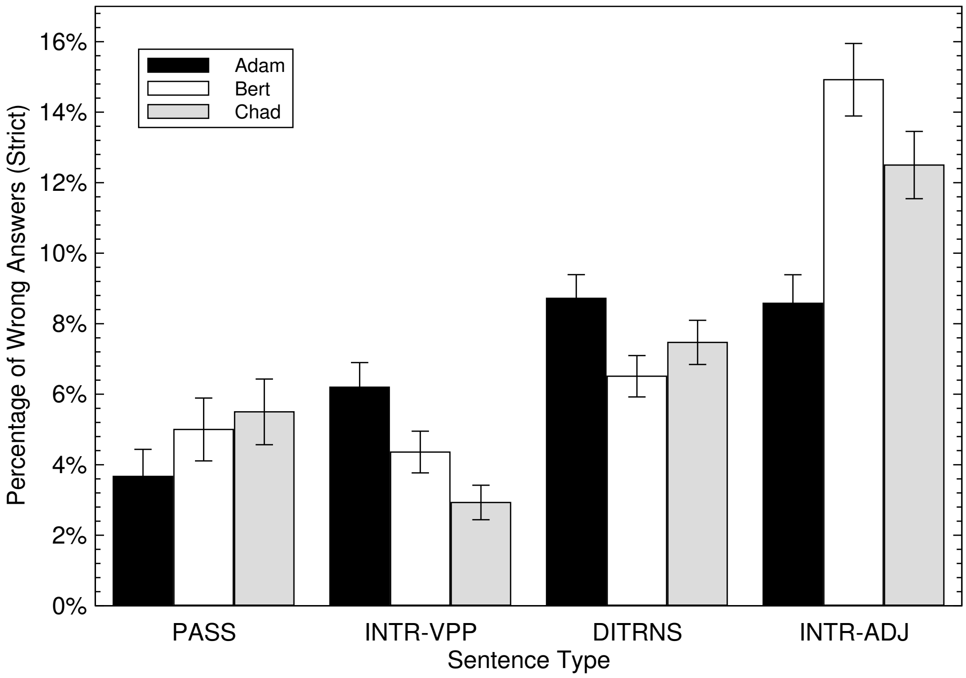


Figure 7.28: Individual differences in comprehension performance on four sentence types.

to distinguish properties of the CSCP model as a whole from idiosyncrasies of particular instantiations. This section takes a look at the extent of individual differences among the networks.

The average performance profiles across training and as a function of the number of propositions, which were shown in Figures 7.1–7.4, would seem to suggest that the three networks have almost identical performance. While it is true that the average comprehension error rates of the models are very similar, that is just a global measure. If the models really had learned to perform quite identically, we might expect the errors they make in responding to individual questions to be highly correlated. However, it turns out that the responses by the networks to the same questions across the testing corpus have an average pairwise correlation of only 0.604 using the strict criterion and 0.494 using the multiple-choice criterion. While these are certainly significant correlations, it is not the case that the models make identical errors. So the extent of individual differences seems to fall somewhere in the middle ground between major differences in overall error rate on the one hand and precise uniformity on the other. This is not at all surprising.

The correlations between the networks' error patterns increase if we group questions together and look at average error rates. If, rather than correlating the individual responses, we look at the average correct response rate per sentence, the strict-criterion correlation increases to 0.888. If we compute individual error rates for the 25 sentence types discussed in Section 7.3, the average correlation among the networks increases to 0.996. Furthermore, the word-by-word reading times of the networks are also very similar, with a correlation of 0.970.

Such high correlations might lead us to believe that the average response rate of one network on a particular sentence type might be very similar to that of another network, and thus representative of the model as a whole. However, this too turns out to be not entirely correct. Correlations tend to be most sensitive to the extreme values. If the average error rates are correlated across only the 16 sentence types with less than 15% error, the correlation between the networks drops from 0.996 to 0.889. Incidentally, there is some difference in the pairwise correlations of the networks. Measured over those 16 sentence types, the correlation between Bert and Chad is 0.945, while that between Bert and Adam is only 0.832.

Figure 7.28 shows the average error rates for the three networks on four of the sentence types. Clearly, at this level there are considerable individual differences among the networks, many of which reach significance. Each network performs better than the others on some sentence types, but worse on other types. The fact that Bert and Chad are more similar to one another than they are to Adam is somewhat evident in the data shown in Figure 7.28. However, in the case of the active intransitives versus passives discussed in Section 7.3.2, it was Adam and Chad who patterned together.

Thus, it is possible for different instances of the model, exposed to identical environments, to develop individual biases or strategies that impact the behavior of the networks both quantitatively and qualitatively. It seems, from looking at individual differences in other data, that as a class of sentences becomes more rare and more homogeneous the variability among the networks in terms of average error rate on those sentences increases. Thus, although the three networks are very similar in their behavior at a broad level, it is not the case that the performance of one network on a particular type of sentence is necessarily a good predictor of the performance of the other networks or the model as a whole. Therefore, it is important to train and test multiple versions of the model, just as we average performance across many human subjects in empirical research. When comparing the model's performance on rare or subtly different sentence types, it may be that just three networks are not sufficient to reliably represent the average instance of the model, but computational limitations prohibit the training of additional networks for the time being.

Chapter 8

The Main Verb/Reduced Relative Ambiguity

Because of the potential difficulties they pose for any parsing mechanism, structural ambiguities have received considerable attention in the psycholinguistic literature. Chapter 3 discussed empirical findings on the difficulty of comprehending three of the most popular temporary structural ambiguities, including the main verb/reduced relative (MV/RR) ambiguity, the reduced sentential complement (NP/S) ambiguity, and the unmarked subordinate clause (NP/0) ambiguity. Structural ambiguities such as these are of particular importance because our ability or failure to comprehend them, and the subtle details of the manner in which we process them, is a principal source of evidence in evaluating models of the human comprehension or parsing system.

Principally, structural ambiguities have provided a battleground between proponents of the Garden Path Theory (Frazier & Fodor, 1978) and those of constraint-satisfaction theories (MacDonald et al., 1994a). Frazier's Garden Path Theory proposes that the human parsing mechanism has two stages, the first of which is sensitive only to syntactic category information. Semantic and pragmatic information plays a role only in the second stage, after an initial interpretation has been selected using purely syntactic information and principles. Constraint-satisfaction theories, on the other hand, maintain that many sources of information, including semantic, pragmatic, and contextual factors, influence even the earliest stages of sentence processing. If sides were to be chosen, the current model would fall well within the constraint-satisfaction tradition.

Perhaps the best-known structural ambiguity is the main verb/reduced relative. The MV/RR ambiguity occurs when a noun is modified by a reduced passive relative clause in the simple past tense whose verb has identical past tense and past participle forms. In the main verb interpretation, the verb is incorrectly read as a past tense main verb. This ambiguity is exemplified by the classic sentence (52a).

(52a) The horse raced past the barn fell.

(52b) The horse that was raced past the barn fell.

(52c) The horse ridden past the barn fell.

There are two main ways in which the MV/RR ambiguity can be avoided. One is to mark the relative clause by not reducing out the relative pronoun and auxiliary verb, as in (52b). The presence or lack of the relative pronoun *that* will be referred to as the marked vs. reduced distinction. Reduced relatives are also unambiguous if the verb has a special past participle form, as in (52c). This will be referred to as the ambiguous vs. unambiguous distinction. Some authors have called the difference between (52a) and (52b) an *ambiguity effect*. To avoid confusion, the term *ambiguity effect* will be reserved for the difference between (52a) and (52c), and the difference between marked and reduced sentences will be referred to as a *reduction effect*.

Most studies of the MV/RR have used sentences of a form similar to (53a) and (53b), where the ambiguous verb is immediately followed by a by-phrase or prepositional phrase which usually serves to disambiguate the sentence. It is important to keep in mind, however, that in Penglish the word *by* is a disambiguator, but in English the by-phrase may be confusable with a locative up to and even following the noun.

(53a) The witness examined by the lawyer confused the jury.

(53b) The evidence examined by the lawyer confused the jury.

8.1 Empirical results

Much of the empirical work on the MV/RR ambiguity has focused on the effect of noun semantics on comprehensibility and reading times. When the subject of the sentence (*witness*) is a good agent of the ambiguous verb (*examined*), as in (53a), such sentences are expected to be more difficult than when the subject is a poor agent, as in (53b). Conversely, if the subject is a good patient the sentence should be easier than if the subject is a poor patient. In theory it may be possible to isolate individual effects of agenthood versus patienthood, crossing both factors. However, the factors tend to be highly correlated. Good agents tend to be poor patients and vice versa. Most experiments that have looked for semantic effects have manipulated either agenthood or patienthood, but not both. For this reason, and because Penglish does not have a large enough vocabulary to completely cross these factors, the distinction will not be clearly drawn here. I suspect that if subjects, or the model, are sensitive to one factor, they are probably sensitive to both factors. If, while reading a certain part of the sentence, subjects are faster on a set of MV/RR sentences with bad agents (or good patients) than they are at sentences with good agents (or bad patients), this will be referred to as a *semantic effect*. If the opposite is true, it will be called a *reverse* semantic effect.

In reviewing some of the major empirical results in the literature, we will begin with experiments testing the effects of semantics and reduction. In referring to conditions, the initials M and R will indicate marked and reduced sentences. A and U will refer to RRs using verbs with ambiguous or unambiguous past participles. If neither is mentioned, A should be assumed. The initials G and B will indicate that the subject is a good or bad agent. It is assumed that good agents are also bad patients, and vice versa. Thus, the condition MB is a marked relative with a bad agent. This is expected to be the easiest condition. RG would be a reduced relative with a good agent, which is expected to be the hardest condition.

The hope in modeling the empirical data to be reviewed here is that the model's behavior in various conditions will match the consistent qualitative patterns of human behavior. Although it would ultimately be desirable to model quantitative data, that is not a reasonable goal for the time being. The principal reason, as we will see, is that different experiments on human subjects do not always agree with each other qualitatively, let alone quantitatively. Subjects' reading times may be significantly affected by the presentation mode, whether it involves eye-tracking, word-by-word or phrase-by-phrase reading, and cumulative or moving window displays. Subjects are also likely to be sensitive to the overall difficulty of the sentences, the types of filler items used, and the difficulty and types of comprehension questions asked. Experiments are also known to differ in the strength of manipulation of factors such as verb frequency or noun plausibility biases. There may also be significant differences between subject pools at various testing sites. How each of these factors impacts reading behavior may be an interesting subject for further study. But until we are sure that the conditions faced by the model are similar to the conditions faced by the human subjects, there will be little point in attempting to match specific numerical results.

Because the CSCP model is not permitted or able to backtrack while reading, its reading time behavior should be compared to first-pass reading times, rather than full or second-pass times in eye-tracking experiments. Therefore, only first-pass reading times are reported in discussing the following experiments. Most experiments report results either as raw reading times, as per-character reading times, or as residuals. It is expected that the per-character or residual reading times will better model the network's behavior because it is not sensitive to word length. Also, self-paced word-by-word reading experiments are a more appropriate test of the model than eye-tracked reading times because the model is not provided with parafoveal access to upcoming words. Because the model does not actually have any control over the amount of time spent on a word, the most appropriate presentation method to use for comparison would probably be rapid serial visual presentation (RSVP). However, that would not give us any reading times, so it is only useful for tests of comprehension or grammatical decision.

Rayner et al. (1983) tested eye-tracked reading times on RG, RB, and MB sentences. Comprehension was tested with a paraphrasing task. Subjects were indeed able to paraphrase the RB sentences better than the RG sentences. However, the semantic effect did not show up in first-pass reading times per character. It is important to note that the materials used in this study were somewhat different from those used in most other studies in that many of the reduced relatives used ditransitive verbs.

Eye-Tracking and Word-by-Word Reading		
Study	*examined*	*by the lawyer*
RCF83	RB $=$ RG $=$ MB	MB $<$ RG $=$ RB
FC86	MB $\leq$ MG $\leq$ RG $<$ RB	MB $=$ MG $<$ RB $\leq$ RG
PM92	RG $<$ MG $=$ MB $<$ RB	MB $\leq$ RB $\leq$ MG $<$ RG
TTG94-1	MB $=$ MG $=$ RG $\leq$ RB	MB $=$ MG $=$ RB $<$ RG
TTG94-2	MB $\leq$ MG $=$ RG $=$ RB	MB $\leq$ MG $\leq$ RB $<$ RG
CSCP Model	MB $\leq$ RG $<$ MG $<$ RB	MB $<$ MG $=$ RG $<$ RB

Two-Word Window Self-Paced Reading			
Study	*examined by*	*the lawyer*	*confused the*
TSMT94	MB $<$ MG $<$ RB $\leq$ RG	MB $<$ RB $<$ MG $<$ RG	MB $<$ RB $=$ MG $<$ RG
BTH94-1	MB $<$ RB MG $<$ RG	MB $<$ RB MG $<$ RG	MB $=$ RB MG $<$ RG
BTH94-2	MB $=$ RB MG $<$ RG	RB $<$ MB MG $<$ RG	MB $<$ RB MG $<$ RG
MFA97	MG $\leq$ MB $<$ RG $\leq$ RB	MG $<$ MB $=$ RB $<$ RG	MB $<$ RB $<$ MG $<$ RG
CSCP Model	MB $<$ RG $\leq$ MG $<$ RB	MB $=$ MG $\leq$ RG $<$ RB	MB $\leq$ RB $<$ MG $\leq$ RG

Table 8.1: Summary of qualitative first-pass reading results on MV/RR sentences. M = marked, R = reduced, G = good agent, B = bad agent, $<$ indicates the condition on the left is probably significantly faster, $\leq$ indicates a smaller difference, probably non-significant, $=$ indicates no appreciable difference.

Table 8.1 shows a summary of the reading-time results for this and the other experiments discussed here. Results are given for both the ambiguous region, which is often just the ambiguous verb, and the disambiguating region, which is generally a by-phrase or other PP. "RB $=$ RG" indicates that reading times for reduced bad subjects were approximately equal to those for reduced good subjects. "MB $<$ RG $=$ RB" means that marked bad subjects were significantly faster at disambiguation than reduced good subjects, which were approximately equal to reduced bad subjects. A $\leq$ relationship is used where two conditions differ marginally, and perhaps not significantly so.

Ferriera and Clifton (1986) fully crossed the M/R and B/G factors in an eye-tracking experiment. Their items used single-object transitives followed by a by-phrase or other PP that usually, but not always, served to disambiguate the sentences. On the verb, they found that the RB condition was slower than the others. The differences among the other conditions were small, but possibly not negligible. At the point of disambiguation, the reduced conditions were slower. There was almost no animacy effect for the marked conditions, but there was a small animacy effect in the reduced condition. Again, these results are summarized in Table 8.1.

Pearlmutter and MacDonald (1992) used a single-word self-paced moving window and compared marked and reduced sentences with subjects that were good or bad themes or patients of the verb. For the purpose of this review, I will treat the good themes as bad agents, although that may not have always been the case. Their items also differ from those in other studies because their by-phrases were not treated as disambiguating, and the disambiguation point was the first two words of the main verb phrase. In the ambiguous region, there was no effect of semantics for marked sentences; but reduced sentences with good agents were faster while those with bad agents were slower than the marked sentences. In the disambiguation region, the RG condition was the slowest, as expected, and the next slowest was the MG condition. Although there was both a reduction and a semantic effect, the semantic effect appears to have been stronger. Furthermore, unlike the Ferriera and Clifton (1986) results, there was still a semantic effect at the disambiguation point for marked sentences. These differences could be due to a stronger semantic manipulation in the Pearlmutter and MacDonald materials.

Trueswell et al. (1994) crossed reduction with noun animacy, which will be treated as agent plausibility, and also added two conditions with unambiguous verbs and bad agents. Reading times were measured with eye-tracking, and rather than using per-character or residual reading times, raw reading times were reported. Unambiguous verbs were read quite a bit faster than ambiguous verbs with or without reduction. There was little other effect on the verb except that RB verbs may have been slightly slower than the other conditions. The fact that unambiguous verbs were read faster than RG verbs suggests that the unambiguous ones were either shorter or more frequent, and a direct comparison is probably not meaningful. On the disambiguating phrase, the RG condition was significantly slower than the other conditions.

In a second experiment, Trueswell et al. (1994) used slightly different materials and eliminated the unambiguous verb condition. The effect on the verb was again quite small but had a slightly different pattern. In this case, the MB condition was numerically slower than the other conditions. At the disambiguating region, the RG condition was again much slower than the others. There may have been slight differences in the other conditions as well, although probably far from significant. Thus, the semantic effect was large only for reduced sentences and the reduction effect was large only for good agents.

Tabossi et al. (1994) used a self-paced two-word moving window design. The ambiguous verb was always paired with the word *by* and was followed in the next window by the agent NP. The subsequent region contained the first two words of the main verb phrase. In the verb+by region, the MB condition was the fastest followed by the MG condition and then the reduced conditions. There was little effect of the semantics for the reduced sentences in this region. Possibly this is because the verb portion of the verb+by region is more difficult in the bad agent case, but the *by* portion is easier and the result is little overall difference. On the NP region there were effects of both semantics and reduction, with the semantic effect being stronger. Interestingly, these effects continued on the next, main verb, region, where the garden path for the RG condition was even stronger.

Burgess et al. (1994) also employed a two-word moving window display and compared the stimuli used by Ferriera and Clifton (1986) with more constraining items. Unfortunately, only reduction effects were reported so we cannot directly compare the good and bad agent conditions. Using the Ferriera and Clifton stimuli, there was a reduction effect on both the verb+by and NP regions for both semantic conditions, but only on the VP region for the good agent condition. Using items with stronger semantic constraints in a second experiment, the reduction effect was eliminated for bad agents on the verb and actually reversed on the NP. It is not clear why this reversal should have occurred.

Rather than using nouns that were themselves good or bad agents of the verb, McRae et al. (1997) pre-modified the nouns with adjectives that affected the plausibility of agenthood. For example, a sentence might begin either, "The shrewd heartless gambler manipulated...," or, "The young naive gambler manipulated..." Presumably, the shrewd, heartless gambler is a more plausible agent of *manipulated*. In the verb+by region, there was a large effect of reduction but little effect of semantics. In the NP region, there was no reduction effect for bad agents, but a large effect for good agents. In the VP region, there were small reduction effects and larger semantic effects. These results are interesting because they show that subjects are not simply sensitive to the frequency of noun/verb relationships in making parsing decisions, but are sensitive to possibly complex, derived semantic representations.

As should be evident in studying Table 8.1, there is far from unanimous agreement in the relationships among these four sentence conditions across experiments. Clearly the results are sensitive to the individual items and/or the methodology used in different experiments. Therefore, it is unreasonable to expect the CSCP, or any model, to closely reproduce the results from any one MV/RR experiment. However, some general trends probably are fairly reliable.

On the ambiguous region in the eye-tracking experiments, there seems to be a small reduction effect and a small reverse ambiguity effect. However, those effects are not always reliable and there is often an apparent interaction such that only one condition differs from the other three. With the exception of Rayner et al. (1983), however, the RB condition is generally the slowest. Using a single-word self-paced moving window display, Pearlmutter and MacDonald (1992) found no ambiguity effect for marked sentences, but a reverse ambiguity effect for reduced ones. Because it does not permit parafoveal viewing of the preposition, this experiment is closest to the condition under which the current model will be tested. On the verb+by region in two-word window displays, there is consistently a reduction effect. However, this is accompanied by a semantic effect in some experiments, but a reverse semantic effect in McRae et al. (1997). Because there may be countervailing effects caused by the verb and the *by*, the existence of a forward or reverse semantic effect, or none at all, is probably sensitive to the individual items or experimental conditions.

In the disambiguating region or regions, there is a fairly consistent finding of a reduction effect accompanied by a semantic effect, the RG condition being the hardest. It seems that the semantic effect is often stronger, but this probably depends on the items. The semantic effect may be stronger for reduced sentences, although it does still appear for marked sentences.

8.2 Experiment 6

An experiment was conducted on the ability of the CSCP model to process MV/RR sentences crossing three factors: whether the RC is marked or reduced, whether the verb has an unambiguous or ambiguous past participle form, and whether the subject is a good or bad agent of the verb. All sentences were of the form shown in (54). Sentences began with an NP consisting of an article (*a* or *the*) and a singular noun. If marked, the RC was introduced by either *that* or *who/which* with equal probability. The by-phrase also contained an NP with an article and a singular noun. The main verb phrase was always in the passive, although the tense varied, and was modified by a PP. There were two reasons for using the passive. One is that some of the subjects were inanimate nouns that could not reasonably serve as subjects of an active sentence. There is simply not much that *evidence* can do given the limited set of verbs in Penglish. The other nice property of passives is that they all start with an auxiliary verb (*is*, *was*, *has*, *had*, or *will*). Therefore, reading times can be compared on these words across conditions with less noise due to verb frequency or plausibility.

(54) The bird [that was] eaten by a cat was given to me.

The semantic manipulation in this experiment was quite strong. Good agents were nouns that could serve as either subjects or objects of the verb. Bad agents were nouns that could not serve as agents of the verb, according to the Penglish grammar. For example, the verb *saw* can have a human or animal as an experiencer, but can have a human, animal, or physical object as a theme. Therefore, the humans and animals are good agents while the physical objects are bad agents.

In some cases there were only a few lexical items to choose from in forming the good and bad agent sets. For example, the verb *ate* can have any human or animal as its agent, but the patient must be *apple*, *food*, *bird*, *fish*, or *something*. Of these, *bird* and *fish* are the only good agents while *apple* and *food* are the only usable bad agents. *Something* cannot be used because it cannot take an article, which was a requirement of all subjects to equate for word position.

Seven unambiguous verbs were selected, including *eaten*, *bitten*, *given*, *known*, *seen*, *shown*, and *taken*. The verbs *thrown* and *written* were eliminated because they have no good agents that are also valid patients. *Forgotten* and *gotten* were eliminated because they never occur in reduced relatives, at least not in Penglish, and *driven* and *flown* were eliminated because they are not productive enough, only allowing *cars* or *planes* as objects, respectively.

Ten ambiguous verbs were used, including *believed*, *felt*, *found*, *followed*, *heard*, *hit*, *killed*, *left*, *examined*, and *wanted*. *Asked* was eliminated because it has no bad agents and *bought*, *read*, *said*, *used*, and *told* because they have no good agents. Also eliminated was *had* because it is ambiguous as an auxiliary, *put* because it requires a PP, *parked* and *played* because they are not productive enough, and *guessed*, *questioned*, and *thought* because they are never reduced.

The unambiguous verbs tend to be more frequent both overall and in passive relative clauses. The average log (natural) frequency of all occurrences is 4.40 for the unambiguous verbs versus 4.18 for the ambiguous ones. The average log frequency of occurrences in passive relative clauses is 2.67 for the unambiguous and 2.31 for the ambiguous. And the log frequency of occurrences in *reduced* passive relative clauses is 2.49 for the unambiguous words and 2.12 for the ambiguous. Therefore, there may be a moderate frequency bias in favor of the unambiguous verbs.

Fifty sentences were generated for each verb in each condition. The marked and reduced sentences were identical, aside from the reduction. In generating sentences, the subject and agent of the RC verb were selected at random from the valid nouns. Depending on the condition, the subjects were either good or bad possible agents of the verb. The sentence predicates, including the passive main verb and its PP modifier, were dependent on the subject noun and were selected from a set of constructions of that form drawn from a large corpus of Penglish. Therefore, the semantic relations expressed in the main clause were all reasonably natural. However, it should be noted that there may be frequency and plausibility differences between the semantic conditions. The poor agents tend to appear as the subject of passive constructions more often than do good agents.

All three networks were tested on all of the experimental items and the results were averaged across the networks. Both comprehension performance and reading time were analyzed.

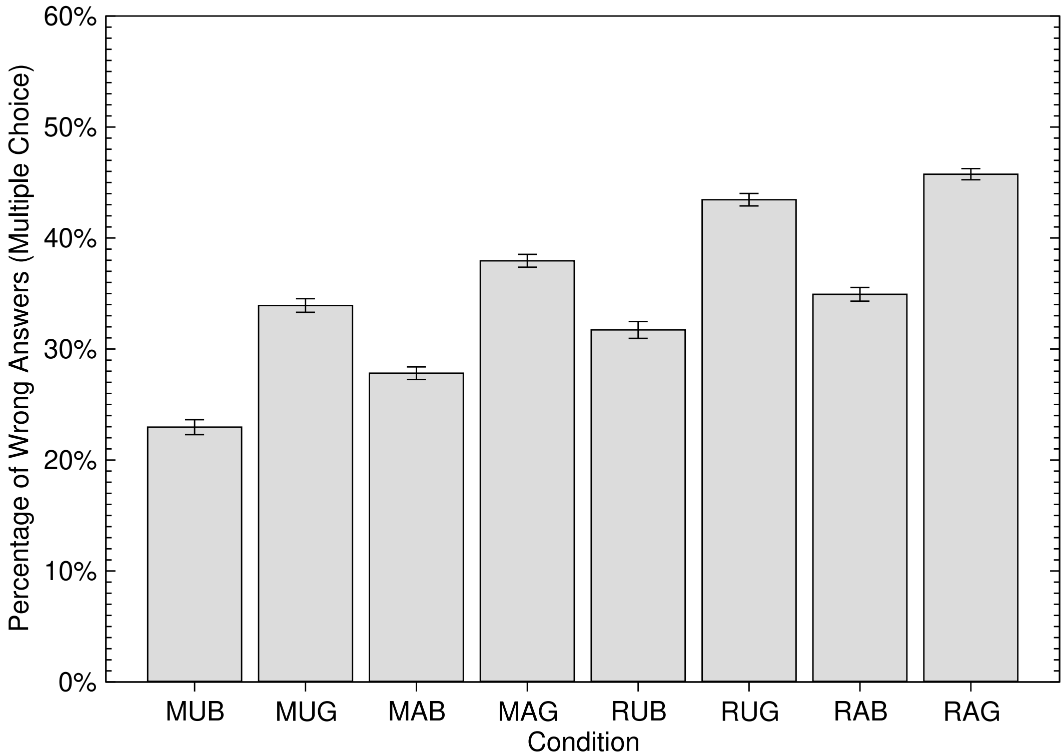


Figure 8.1: Multiple-choice comprehension error across all conditions in Experiment 6.

8.2.1 Comprehension results

We begin by looking at the comprehension performance of the model. Few of the empirical studies of the MV/RR reported their comprehension results, and some of the studies asked comprehension questions only sporadically. Therefore, there is not a good basis for comparison of these results. However, the model's results should serve as predictions for future tests of human comprehension performance under similar conditions.

Figure 8.1 shows the average multiple-choice error across the eight conditions. As expected, marked sentences are easier to comprehend than reduced ones, unambiguous verbs are easier than ambiguous ones, and subjects that are bad agents are easier than good agents. Figure 8.2 shows the three main effects of these factors, all of which are highly significant ($p < 0.001$). The effect of verb ambiguity is the smallest, while the effect of semantics is the largest. There is also a significant interaction between ambiguity and reduction;[1] reduction actually has a larger effect for unambiguous verbs, which is a bit counterintuitive. The other two-way interactions and the three-way interaction are not significant. Therefore, the model shows clear sensitivity to the three factors in its comprehension performance.

We can get a better understanding of the types of comprehension errors made by the network by breaking down the errors based on the proposition being queried. Figure 8.3 shows the error rate for each of the four propositions in the sentence as a function of the condition. Only sentences with ambiguous verbs are included. For each proposition, error rates for the three possible questions are averaged. The *MV Patient* proposition expresses the fact that the subject is the theme or patient of the main verb. The *RC Patient* proposition expresses the relationship between the subject and the RC verb. The *RC Agent* is the relationship between the RC verb and the object of the by-phrase. Finally, the *MV PP Object* proposition encodes the relationship between the main verb and the object of its prepositional modifier.

The model is very good at the prepositional phrases. Not only do they occur at the end of the sentence, prepositional phrases are quite frequent in Penglish and, in this case, are unambiguous. The model has moderate difficulty with the *MV Patient* and *RC Agent* relationships. However, error is considerably worse on the relationship between the subject and the relative clause verb. This is especially true of the reduced good agent condition, which has an average of 84%

[1] F(1,848)=9.67, p=0.002

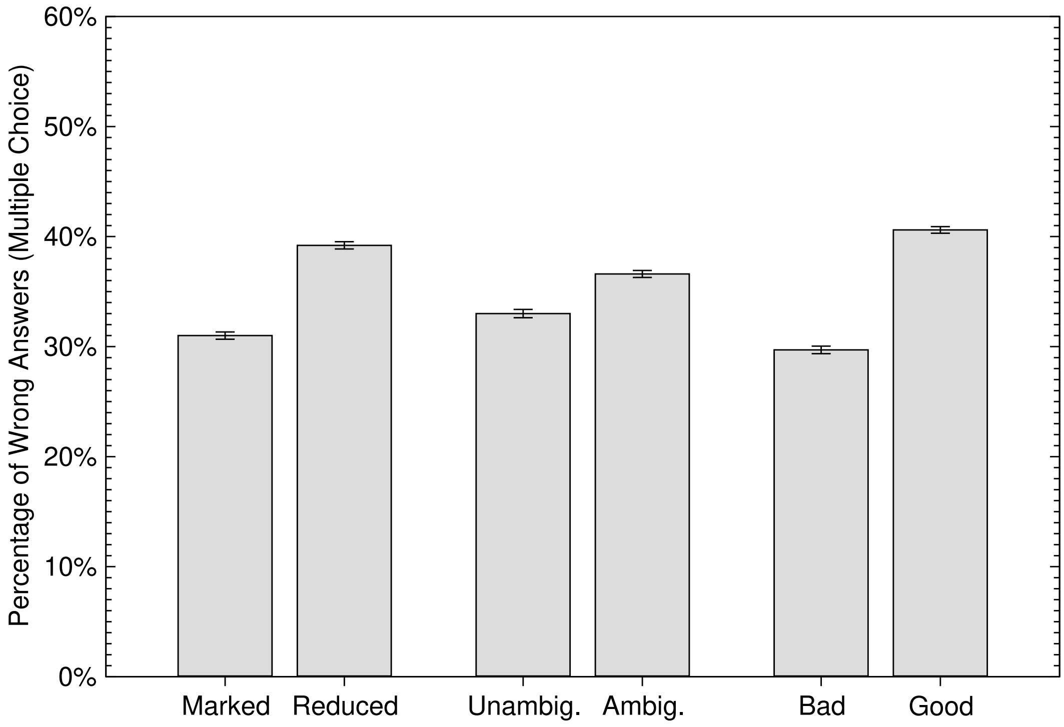


Figure 8.2: Main effects of the three factors on multiple-choice comprehension error.

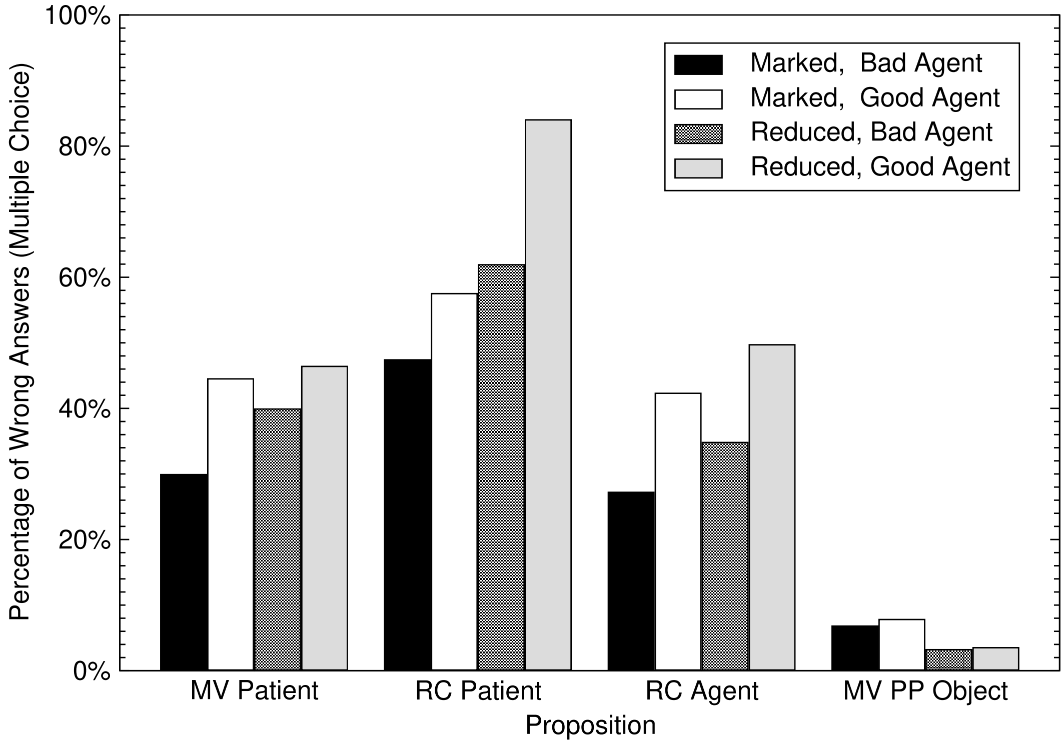


Figure 8.3: Comprehension errors on each of the four propositions for ambiguous verb sentences.

error. If given a sentence like (53a), which is reduced and has a good agent, and then asked, “What is the relationship between *examined* and *witness*,” the model gets the wrong answer 98.6% of the time. Therefore, it is clear that the model is almost never recovering from the garden path in this condition. In the marked condition with a bad agent, the error on this question drops to 65%. This is far from perfect, but the model is at least able to comprehend such sentences to some degree.

It is not known how well humans would perform in answering a similar multiple-choice question. However, MacDonald et al. (1992) tested comprehension performance on main verb and reduced relative sentences using true/false questions. Although error rates were around 10% for MV sentences, high-reading-span subjects made about 33% errors on RR sentences and low-span subjects made 48% errors, which is essentially at chance. Therefore, although the model has difficulty comprehending RR sentences, it is clear that humans do as well. In order to properly evaluate the model, and to better understand human performance, it will be necessary to more carefully measure comprehension of MV/RR sentences using a variety of questions focused on specific aspects of sentence meaning. Multiple-choice questions will probably be more revealing, and are better suited for comparison with the model, than are true/false questions.

8.2.2 Reading time results

Given the available behavioral data, the more interesting measure than comprehension is reading time. Reading times were assessed using the SRT measure, which was explained in Sections 6.5.4 and 7.7. The SRT is sensitive both to prediction errors and to the degree and difficulty of message change. It gives word-by-word reading times that could be thought of as residuals, as there is no effect of word length.

Figure 8.4 shows the SRT for each word in the marked sentences. The letters appearing above the curve provide a quick reference to the factors that apparently result in slower reading at that point. B indicates that bad agents are harder than good agents at that point. A G indicates that both ambiguous verbs and good agents contribute to the difficulty. On the subject noun, *evidence*, bad agents are read slower than good agents. This is presumably because bad agents are less frequent in the subject position and thus unexpected and harder to process. The semantic effect continues on *that*. The reason for this is less clear, it appears that both the prediction and semantic change portions of the SRT are slower on *that* for bad agents. It may be that bad agents are less likely to be modified by relative clauses and the model is sensitive to this. Or it may be the fact that the bad agents are unusual in subject position that carries over to the next word.

At the auxiliary verb, the semantic effect reverses. This is presumably because the auxiliary verb indicates that a passive construction is likely (although not guaranteed) and good agents are less likely to be the subject of a passive. At the RC verb, there are effects of both ambiguity and semantics which appear to be relatively independent. Good agents are harder, as are ambiguous verbs. It is interesting that ambiguous verbs are more difficult here. It is possible that this is a frequency effect, but other factors may be at work as well. Perhaps there is added work required to disambiguate the form of ambiguous verbs, but this could only be verified if frequency were better controlled for. At the disambiguation point, there is possibly a small effect of ambiguity, but there is definitely a larger semantic effect which goes in the expected direction.

Figure 8.5 shows similar reading-time results for the reduced sentences. SRT on the subject is identical to the marked case. The interesting effect occurs on the RC verb. The faster condition is the ambiguous verb with the good agent, presumably because the model is treating it as a main verb. The slowest condition by far is the unambiguous verb with a bad agent. The model clearly must be treating this as a reduced relative. The larger reading time is partially due to the fact that a past participle was unexpected. But another contributing factor is the larger change in semantic representation that is required to encode the fact that a relative clause is occurring. The AB and UG conditions fall between the other two in terms of reading rate. Presumably, in those cases, the model is somewhere between fully adopting a MV or RR reading.

Reading times on the word *by* are somewhat perplexing. We should expect the good agents to result in a garden path effect here. However, although the effects are small, the bad agent condition is actually slower at disambiguation. Thus, although the model showed a proper semantic effect on *by* in the marked sentences, it is showing a reverse ambiguity effect on the reduced sentences. This is a clear mismatch between the model's performance and human behavior. Interestingly, this reverse ambiguity effect carries over to the article, whereas there were no effects on the

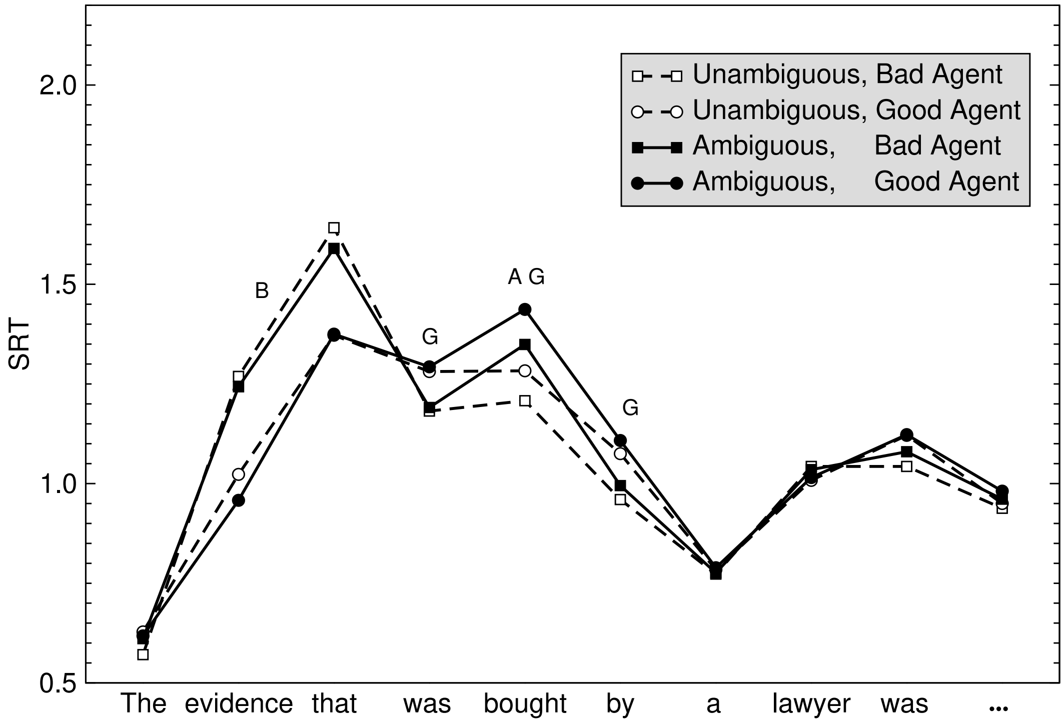


Figure 8.4: Reading times for the marked MV/RR sentences in Experiment 6.

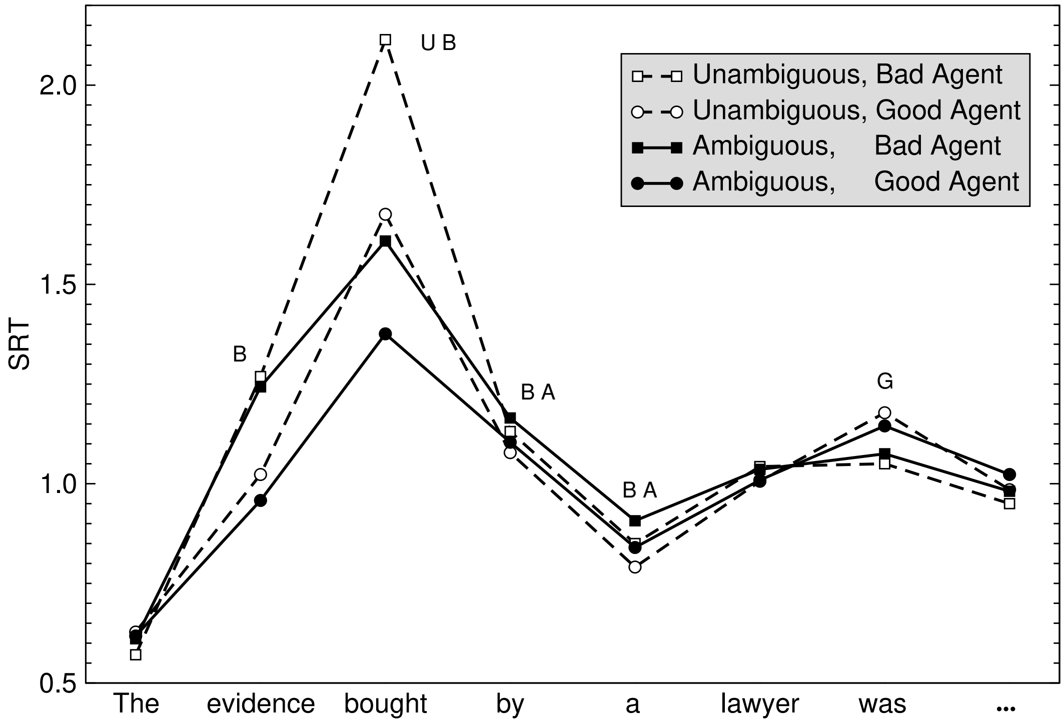


Figure 8.5: Reading times for the reduced MV/RR sentences in Experiment 6.

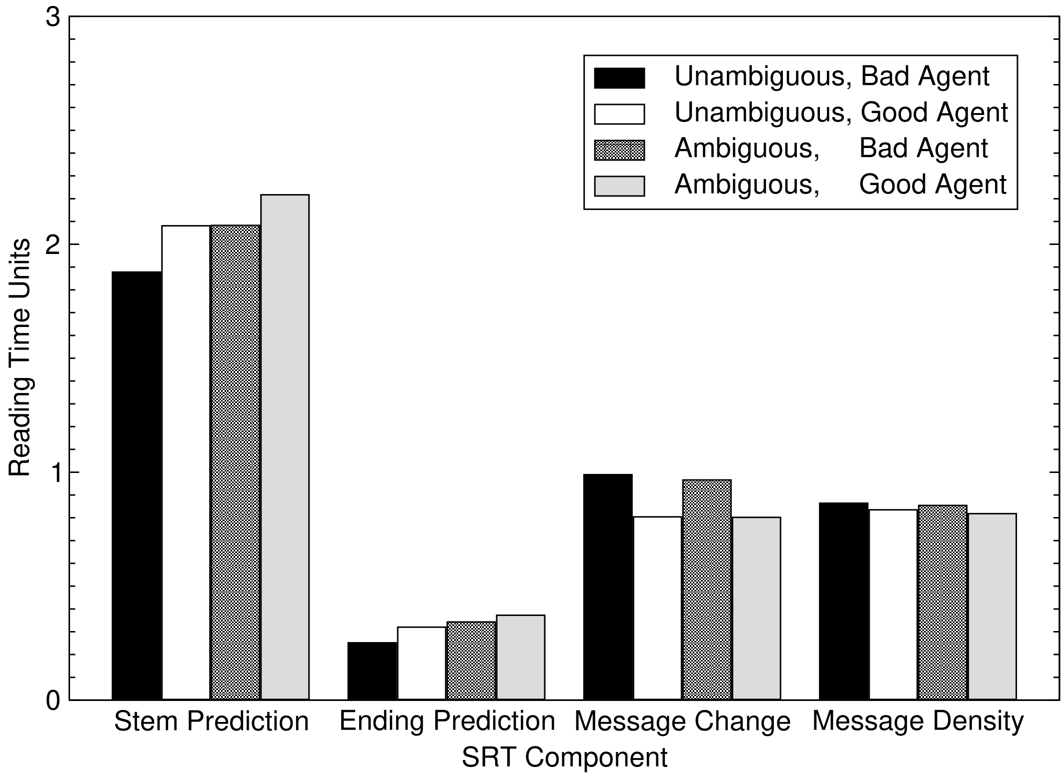


Figure 8.6: Components of the SRT in reading the word *by* in reduced sentences in Experiment 6.

article in the marked case.

One explanation for all of this may be that, in the reduced case with a good agent, the model is not fully recovering from the garden path. Human readers presumably realize that something is wrong when they have garden-pathed and pause to reanalyze the sentence before continuing. In an eye-tracking experiment, this reanalysis may often involve actual re-reading of earlier parts of the sentence. In moving-window experiments that is not possible. However, readers probably store a phonological encoding of the sentence in a short-term memory buffer and may perform reanalysis by replaying part of the sentence from such a buffer. Reanalysis of this sort would result in a significant pause, but should improve comprehension performance. Unfortunately, the CSCP model has been provided with no mechanism for engaging in reanalysis when it is confused. Therefore, it may simply be coasting along in blissful confusion in the difficult good agent condition. The bad agent condition may be slower because the model is at least performing some useful work, which is reflected in the significantly better comprehension performance.

A closer look at the individual SRT components on reading *by* provides some evidence in support of this explanation for the reverse semantic effect. Figure 8.6 shows the normalized values of the four components of the SRT on the word *by* in the four reduced conditions. The two prediction components, predicting the stem and ending of the word, show both ambiguity and semantic effects, in the expected direction. *By* was indeed more surprising for the good agents. However, the message change and, to some extent, the message density components show reverse semantic effects. There is a greater change in the message for the bad agents. Because the message components account for the majority of the SRT in the current weighting, the net effect is a small reverse ambiguity effect. These results are consistent with the theory that the model is essentially giving up on comprehending the difficult good agent sentences.

So far we have not directly compared the marked and reduced conditions. To put all eight conditions on the same graph would be overwhelming, therefore we will first compare the marked and reduced sentences by collapsing across the semantic condition, as shown in Figure 8.7. There are no significant effects until the RC verb is reached. But at that point things get interesting. For the ambiguous verbs there is only a small effect of reduction. But for the unambiguous verbs there is a large reduction effect. An unambiguous past participle following the subject is difficult for two reasons. It is unexpected, but it also clearly marks that a reduced relative is occurring, which requires a large

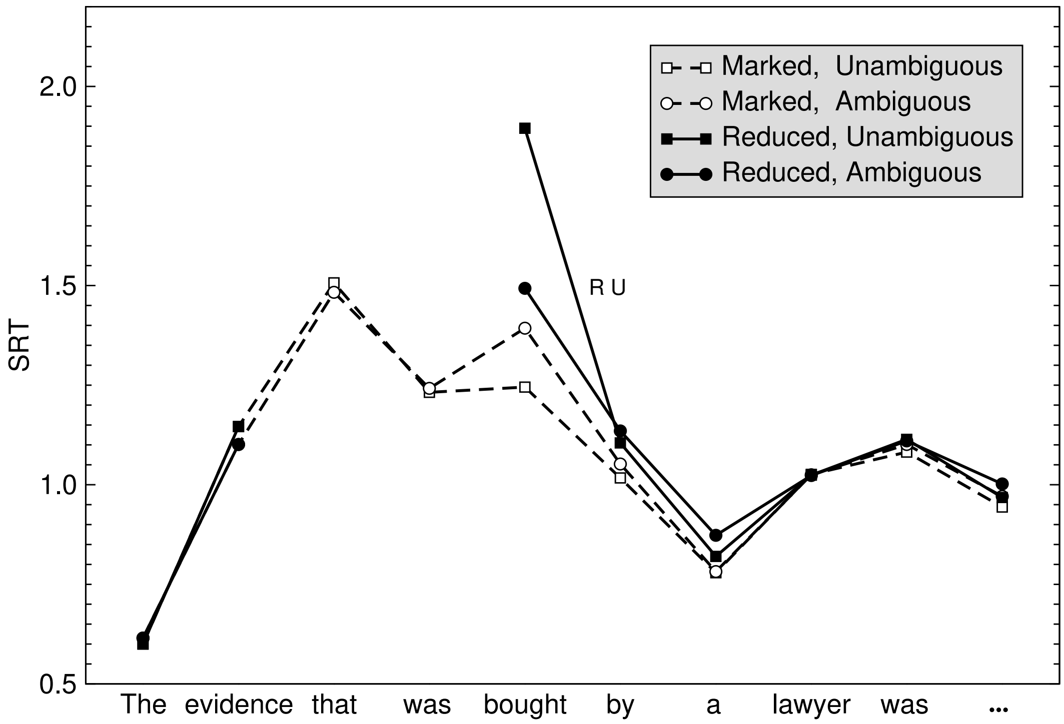


Figure 8.7: Reading times for the ambiguity and reduction conditions in Experiment 6, collapsing across the semantic factor.

change in the semantic representation. Both the prediction and semantic components of the SRT will be elevated.

The fact that the marked conditions are easier than the reduced ambiguous conditions is somewhat surprising. One reason may be that the reduced ambiguous verbs are slow in the bad agent case because they are infelicitous as main verbs and the model must entertain both the MV and RR interpretations. Another factor may be that the marked RC verbs follow an auxiliary verb. Some of the work required to build the semantic and/or syntactic representation of the relative clause has already been done in processing the auxiliary. Therefore, less work in required to add the RC verb. On the word *by* and the following article, there is a small reduction effect and an even smaller ambiguity effect. Although these effects are in the expected direction, they are not large.

Figure 8.8 shows the reading time results for the four ambiguous verb conditions. Rather than collapsing across the ambiguous and unambiguous conditions, the figure shows only the ambiguous ones because most of the empirical studies of semantic and reduction effects only include ambiguous verbs. On the noun, and on *that* in the marked condition, we again see the advantage for good agents, which reverses when the auxiliary verb is reached. On the RC verb, on *by*, and on the article, there is an interaction between reduction and noun semantics. In the marked case, the good agents are harder. Presumably this is because the model is correctly interpreting this as a passive relative clause, but the good agents are not as felicitous as the subject of a passive construction. However, in the reduced case the effect reverses and the bad objects are slower. As discussed earlier, this may be because the model is not properly comprehending the reduced relative with a good agent subject, and is therefore faster, albeit confused.

In order to facilitate comparison with the empirical results, Figures 8.9 and 8.10 show the model's reading times averaged across regions. Figure 8.9 breaks down the results as is typically done in eye-tracking or word-by-word reading experiments, while Figure 8.10 partitions the results into two-word windows. In the verb region, *bought*, and in the verb+by region, there is an interaction between semantics and reduction, with good agents slower in the marked case and bad agents in the reduced case. The last row of each section of Table 8.1 gives a summary of the model's reading time behavior on the MV/RR task.

The fact that the MB condition is the fastest and the RB condition is the slowest is the most consistent finding in the empirical literature and is also produced by the model. However, the RG condition in the model was slightly faster on

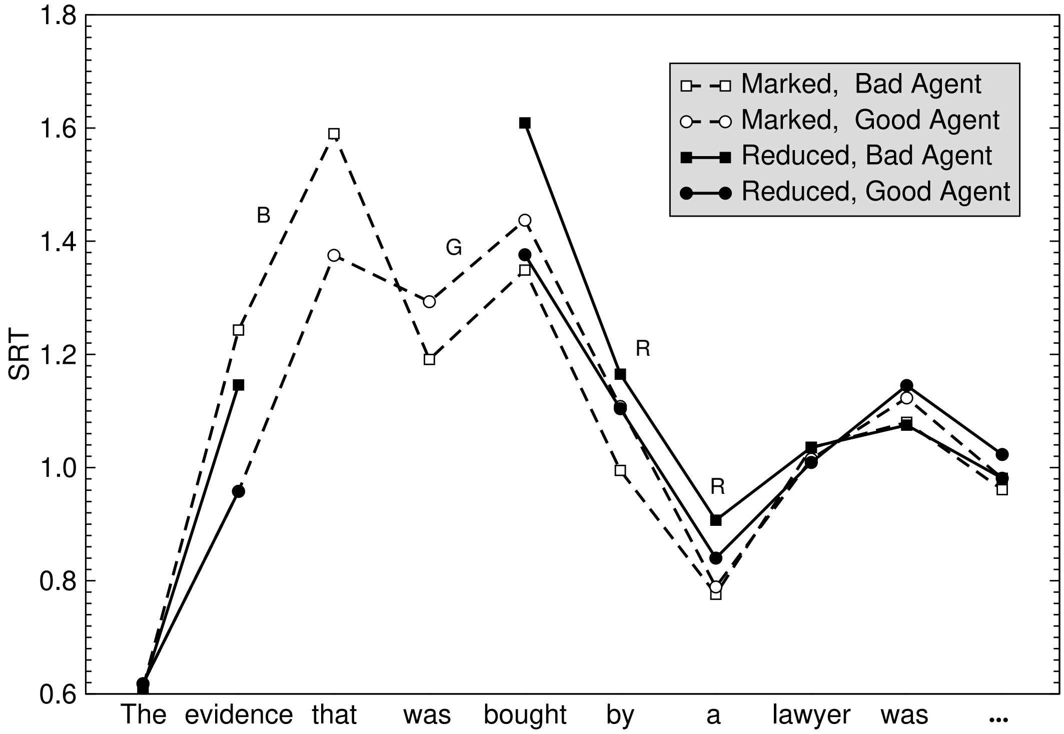


Figure 8.8: Word-by-word reading times for ambiguous verbs in Experiment 6.

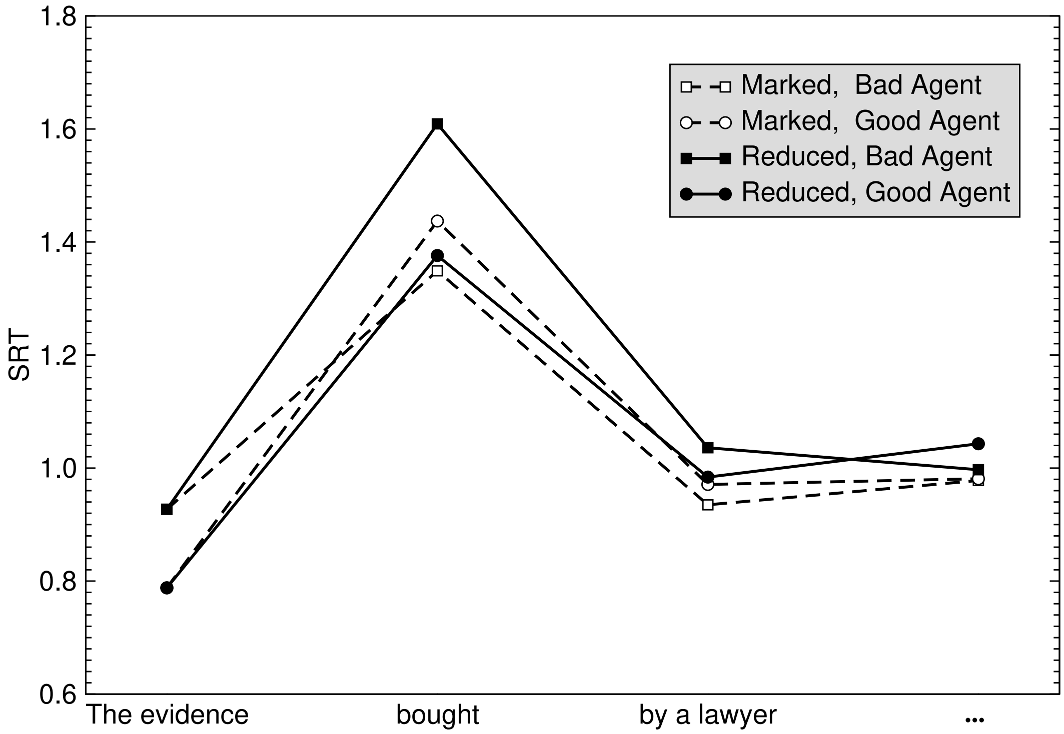


Figure 8.9: Reading times for ambiguous verbs in Experiment 6, averaged across typical regions of interest.

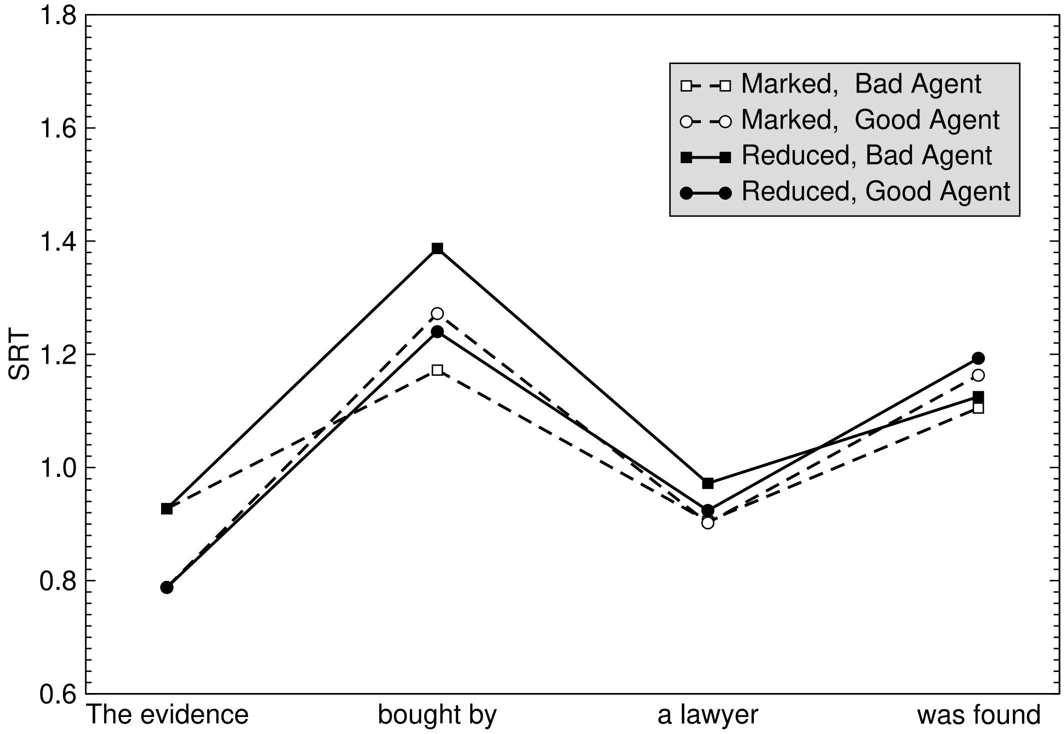


Figure 8.10: Reading times for ambiguous verbs in Experiment 6, averaged across two-word windows.

the verb than was the MG condition. Most of the empirical studies find little difference or a slight advantage for the MG condition, but these may be insignificant.

On the by-phrase, or on the NP when using a two-word window, the model finds the MB condition to be the easiest and RB to be the hardest. The empirical results also find the MB condition to be the easiest, but they generally find RG to be the hardest. That is, the empirical studies find plausibility effects for both marked and reduced sentences, which tend to be stronger for the reduced sentences, but the current model shows a plausibility effect for marked sentences but a reverse plausibility effect for reduced sentences. I have suggested that this is due to the model failing to recover from the garden-path in the difficult RG condition.

8.3 Verb frequency effects

So far we have focused on the effects of reduction, verb ambiguity, and semantics. Some studies have also found evidence that human readers are sensitive to verb-argument-structure frequencies in resolving the MV/RR ambiguity. MacDonald (1994) compared RR sentences with verbs that are always transitive with those with optionally intransitive verbs. The expectation was that the RR interpretation would be strengthened in the obligatorily transitive verbs once the presence of a direct object had been ruled out.

Indeed, the results indicated that optional intransitive verbs led to faster reading times on the RC verb and the following phrase, but longer reading times on the main verb and, possibly, in the region following. MacDonald included an additional factor, which was whether the phrase following the RC verb immediately rules out the MV interpretation or is consistent with it for one or more words.

Unfortunately, Penglish is not quite flexible enough to include such a condition. However, we can test the effect of argument-structure frequencies by reanalyzing the results from Experiment 6. In doing this, we will focus on just the ambiguous verbs and only consider the reduced relatives, averaging over the good and bad agent conditions. The frequencies of the 10 ambiguous verbs appearing in transitive form are shown in Table 8.2. Four of the verbs are obligatorily transitive and six are optionally intransitive, although all but one are biased toward the transitive reading.

Verb	Trans.	Intrans.	% Trans.
felt	5,131	0	100%
found	24,663	0	100%
examined	2,118	0	100%
wanted	5,936	0	100%
killed	11,288	1,410	88.9%
hit	7,700	1,364	85.0%
heard	9,291	3,701	71.5%
followed	8,434	3,668	69.7%
left	16,896	7,583	69.0%
believed	520	1,270	29.1%

Table 8.2: Frequency of verb use in transitive and intransitive forms per 1 million Penglish sentences.

Figure 8.11 shows the reading times for the CSCP model on various regions of the RR sentences, with the results broken down by verb type. As in the MacDonald (1994) findings, the optionally intransitive conditions are faster on the RC verb and the by-phrase, but slower on the main verb and the following regions. Although the differences are not large on all but the first region, they are significant.[2] Furthermore, the model has better comprehension performance for the transitive-only verbs than for the optionally intransitive verbs (37.9% vs. 41.9% multiple-choice error).[3] These results provide some evidence that the model, like humans, is sensitive to verb-argument-structure frequencies in resolving RR ambiguities. However, it is important to note that, with only four and six verbs per condition, it is possible that these results are overly dependent on the individual properties of those verbs and may not actually reflect sensitivity to verb-argument-structure preference itself.

Another verb frequency factor that has been shown to have an effect on the MV/RR ambiguity is the relative frequency of past participle and past tense uses of the RC verb. MacDonald et al. (1994a) found, in a reanalysis of the Pearlmutter and MacDonald (1992) experiment, that sentences using verbs with more frequent use of the past participle than the past tense had higher plausibility ratings. Trueswell (1996) also showed, through meta-analyses, that relative past participle frequency had a negative correlation with reading time and that reading times for reduced relative verbs with high relative past participle frequencies were no slower than those for marked RCs, provided noun semantics favored the RR interpretation.

We should, of course, hope that the CSCP model shows similar sensitivity to past participle frequency. In order to test this, the frequency of the ten ambiguous verbs occurring in active past-tense forms was compared to their frequency in passive forms using the past participle. The frequency counts, out of 1 million Penglish sentences, and the proportion of past participle uses are shown in Table 8.3. Unfortunately, there is not much variation in past participle frequency among these verbs. None of the verbs has a majority of passive uses. Two of the verbs, *felt* and *wanted* have around 18% passive use, while the rest fall in the range 29.9% to 36.9%. Given the small number of verbs and limited frequency variation, it would not be worthwhile to look for effects.

8.4 Summary

In comprehending reduced relative sentences, the CSCP model was able to replicate many of the behavioral effects shown by humans, and also produced some behaviors not yet tested in humans. In its comprehension performance, the model is sensitive to all three manipulated factors: reduction, verb form ambiguity, and noun agency. In addition, a meta-analysis showed that the model had better comprehension performance on the obligatorily transitive RC verbs than on the optionally intransitive ones. The comprehension questions that are most confusing for the model are those regarding whether the subject is the patient/theme of the RC verb, and the next most confusing proposition is that the subject is the patient/theme of the passive main verb. The model performed particularly poorly in answering these

[2] bought: $F(1,2998)=522$, $p<0.001$; by a lawyer: $F(1,8998)=95.3$, $p<0.001$; was found: $F(1,5998)=9.54$, $p=0.002$; ending: $F(1,11803)=18.32$, $p<0.001$

[3] $F(1,71662)=117.7$, $p<0.001$

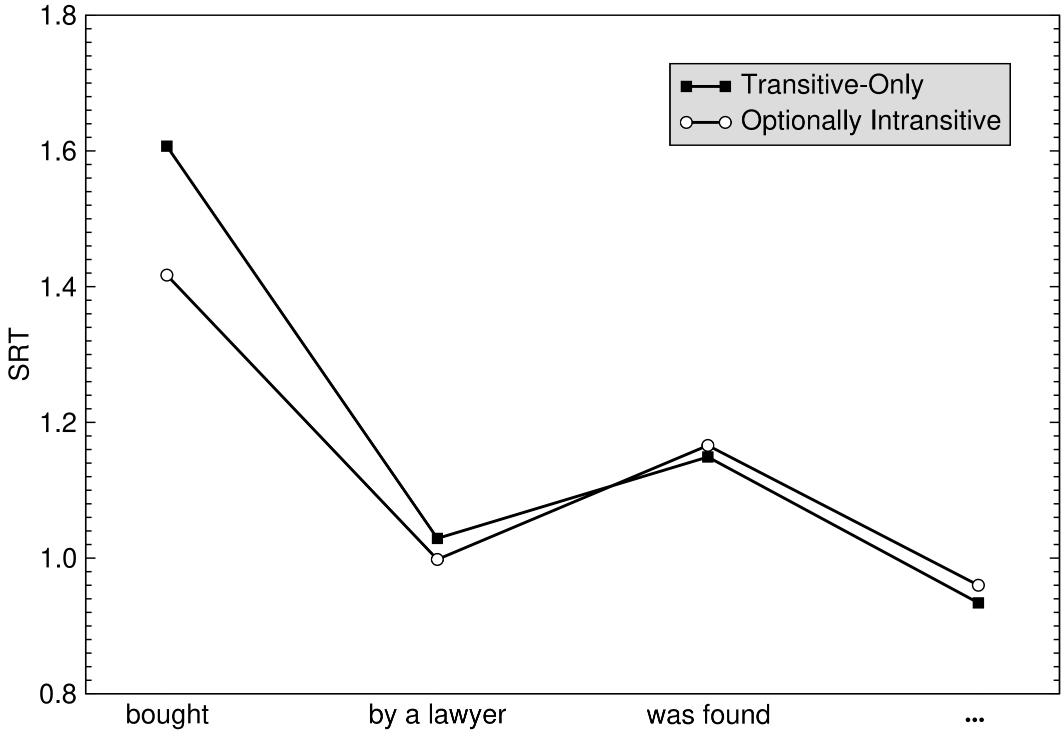


Figure 8.11: Reading times for ambiguous verbs that are either transitive or optionally intransitive.

Verb	Active	Passive	% Passive
followed	12,089	7,080	36.9%
believed	6,233	3,301	34.6%
left	24,452	12,669	34.1%
found	35,164	16,874	32.4%
killed	12,695	5,770	31.2%
examined	2,118	921	30.3%
hit	9,061	3,894	30.1%
heard	14,298	6,097	29.9%
wanted	5,936	1,308	18.1%
felt	11,804	2,592	18.0%

Table 8.3: Frequency of verb use in active past tense and passive past participle forms per 1 million Penglish sentences.

questions in the conditions most biased in favor of the incorrect MV interpretation.

In terms of reading times, the model predicts that unambiguous verbs will be slower on the verb when reduced, but faster when marked. At the disambiguation, sentences with unambiguous verbs are somewhat slower. Similarly, bad agents result in slower reading of the verb when reduced, but faster reading when marked. However, these findings appear to continue at the disambiguation point, where we should expect that bad agents would result in faster reading than good agents. This is perhaps the most problematic mismatch between the behavior of humans and the model. It seems likely that this effect is due to the model's failing to recover from the garden-path in the reduced good agent condition. As a result, it does not attempt to restructure its semantic representations in this condition so that reading times at disambiguation are faster than they should be.

Marking of passive relative clauses tends to result in faster reading of the verb in conditions that favor the RC interpretation, but similar or slower reading times in conditions that favor the MV interpretation. Reduced sentences are slower at the disambiguation point. Finally, as found by MacDonald (1994), optionally intransitive verbs are faster on the verb and on the by-phrase but slower on subsequent regions.

A general problem with the model's performance is that the effects of certain factors on reading time at the point of disambiguation tend to be quite small, although their effects on comprehension performance may be quite large. It is possible that the somewhat impoverished method of extracting reading times from the model simply does not reflect all of the processes actually present in the model that ought to contribute to slower reading. But it is also likely that this reflects a fundamental limitation in the way the model resolves ambiguities. The long reading times of subjects at disambiguation, under conditions that favor the MV interpretation, most likely reflect reanalysis. The readers could be pausing to replay and reanalyze an earlier part of the sentence, either using a phonological buffer or, if possible, by re-reading. On average, this results in considerably longer reading times at disambiguation under difficult conditions. The model, on the other hand, has no mechanism for triggering reanalysis when it becomes confused. Therefore, it attempts to forge ahead as best it can. In very difficult conditions, the amount of processing performed, and thus its reading times, may actually decrease until the model gets back on familiar territory.

In order to test this explanation, it would seem necessary to develop a diagnostic that would allow us to distinguish situations in which human readers are reanalyzing, which in the terminology used here involves replaying the phonological form of the words in a sentence from an internal buffer, and situations in which they are simply repairing, which, by hypothesis, involves restructuring higher-level representations of syntax or semantics without revisiting earlier surface forms. The CSCP model, in its present form, can only be expected to account for restructuring. But the current model aside, any complete account of sentence comprehension must explain the situations in which reanalysis does and does not occur, and the impact this has on reading time and comprehension ability.

Unfortunately, it seems that isolating restructuring from reanalysis may be extremely difficult. One approach might be to use imaging techniques, perhaps EEG or MEG, to identify a signature for reanalysis. Such a signature would have to be quite reliable before it is of any use, which raises the question of how one could independently verify that reanalysis had occurred. An alternative, or perhaps complementary, approach is to find a way to selectively block reanalysis, possibly by interfering with any phonological short-term memory mechanism on which it relies. This might be possible through non-invasive means by giving the reader a concurrent phonological memory task. But one would have to demonstrate that such a task does not interfere with ordinary first-pass sentence processing, which is probably unlikely to be the case. Reanalysis will be an important area for future research, and we may not be able to fully understand first-pass comprehension in the face of ambiguity unless it can be distinguished from reanalysis.

Chapter 9

The Sentential Complement Ambiguity

Another frequently studied structure in English is the sentential complement, or NP/S, ambiguity. The sentential complement ambiguity occurs when a verb that can take either a sentential complement or a noun complement is followed by a reduced SC, as in (55a). The subject of the SC is temporarily ambiguous as an object of the main verb. As with the MV/RR ambiguity, several factors are thought to influence the difficulty of SC sentences. These include whether the SC is marked or reduced, whether the verb occurs more often with SC or NP complements, and whether the NP following the verb is a plausible object. We begin with a brief review of empirical findings.

(55a) The historian read [that] the manuscript of his book had been lost.

(55b) The historian suspected [that] the manuscript of his book had been lost.

9.1 Empirical results

Mitchell and Holmes (1985) tested phrase-by-phrase reading times of marked and reduced sentences like (55a) and (55b) using either *NP-biased* verbs (*read*) that favored the NP reading, or *SC-biased* verbs (*suspected*), that favored the SC reading. The NP was sometimes simple, but in other cases was modified by an adjective, PP, or RC. The only region for which reading times were reported was the disambiguation region, *had been lost*. When the SC was reduced, subjects were faster in the SC-biased case. But when the SC was marked with the complementizer *that*, subjects had approximately equal reading times in the two cases. In comparison to other studies, however, this one used relatively few items and materials that were not well controlled.

The results of the relevant studies are summarized in Table 9.1. The symbols M and R refer to marked and reduced SCs. S and N denote SC-biased and NP-biased verbs. G and B indicate whether the subject of the SC was a good or bad object of the verb. A good object is expected to be harder because it encourages the NP reading. Although noun semantics was not manipulated as a factor, the Mitchell and Holmes (1985) items appeared to use mainly good objects.

Holmes et al. (1987) compared marked and reduced SCs using cumulative word-by-word self-paced reading. Here too the ambiguous NPs varied in form, often containing a PP or RC. The authors found, in the disambiguating region, that marked sentences were slightly, but not significantly, faster than reduced ones. An additional condition was included in which the ambiguous NP was actually the direct object of the verb. However, the disambiguating regions differed in this case and it does not seem to be reasonable to make a direct comparison. Other studies that tried to make direct comparisons between reading times on non-matching phrases have generally been eliminated from this review. Holmes (1987) performed a meta-analysis of the results from this study, classifying the verbs as either NP- or SC-biased. This is the basis for the results shown in Table 9.1. The NP-biased verbs, with reduced SCs, resulted in significantly longer reading times at the disambiguation point than the SC-biased verbs or either of the marked conditions. For the SC-biased verbs, reduced sentences actually had faster reading times at disambiguation than marked sentences, although the difference was not significant.

Rayner and Frazier (1987) tested similar, though not identical, items as Holmes et al. (1987), but measured reading times using eye-movements. They found that reading times in the disambiguation region were significantly shorter for

Study	**Method**	**Ambiguous Region (*the manuscript [of his book]*)**
HKM87	CSP	RS = MN = MS $\leq$ RN
H87	CSP	MN $\leq$ RN, MS $\leq$ RS
HSC89-1	CSP	MNG $\leq * \leq$ RB
HSC89-2	SP	RS = RNG = MNB $<$ MS = RNB = MNG
HSC89-3	SP	S $\leq$ N
TTK93-2	SP	MN = MS $<$ RN $<$ RS
TTK93-3	ET	MN = RN $<$ MS = RS
GPML97-1	ET	MNB $<$ MSG = MSB $\leq$ MNG $\leq$ RNG $\leq$ RSB $\leq$ RSG $\leq$ RNB
GPML97-2	SP	MSG = MNG = MSB $\leq$ MNB $<$ RSG $\leq$ RSB $\leq$ RNB $\leq$ RNG
PTC00	ET	RSG $<$ RSB
CSCP Model	SP?	MNG $\leq$ RNG = MSG $<$ MSB = RSG $<$ MNB $\leq$ RNB $\leq$ RSB
Study	**Method**	**Disambiguating Region (*had been [lost]*)**
MH85	SP	RSG $<$ MSG = MNG $<$ RNG
HKM87	CSP	MN = RS $\leq$ MS $<$ RN
RF87	ET	M $<$ R
H87	CSP	MS $\leq$ MN $<$ RS $<$ RN
HSC89-1	CSP	MS $\leq$ MN $<$ RSB $\leq$ RSG $<$ RNB $\leq$ RNG
HSC89-2	SP	S = MNB $<$ MNG $\leq$ RNG $<$ RNB
HSC89-3	SP	MS = RSH $\leq$ MN $<$ RL $<$ RNH
TTK93-2	SP	MN $\leq$ MS $<$ RS $<$ RN
TTK93-3	ET	MS $<$ MN $<$ RS $<$ RN
GPML97-1	ET	MNB $\leq$ MSG = RSG $\leq$ MNG = RSB $\leq$ MSB $\leq$ RNB $<$ RNG
GPML97-2	SP	RSG = MSG = MNB = MSB $\leq$ MNG $\leq$ RSB $<$ RNB $<$ RNG
PTC00	ET	RSB $\leq$ RSG
CSCP Model	SP?	MNB $<$ MSB $\leq$ MSG $\leq$ MNG $<$ RSB $<$ RSG $<$ RNB $\leq$ RNG

Table 9.1: Summary of qualitative first pass or self-paced reading results on SC sentences. Methods: SP = self-paced reading, CSP = cumulative display self-paced, ET = eye-tracking. Conditions: M = marked, R = reduced, S = SC-biased verb, N = NP-biased verb, G = good object, B = bad object, L = long NP, H = short NP, $*$ = all other conditions. Relations: $<$ indicates the condition on the left is probably significantly faster, $\leq$ indicates a smaller difference, probably non-significant, $=$ indicates no appreciable difference. The experiments differ in the length of the ambiguous region and what they consider the disambiguation.

marked SCs. Reduced SCs also had a greater rate of regression to earlier regions. No results seem to be available for different verb classes or regions within the sentence.

In addition to the meta-analysis, Holmes (1987) performed a direct study of the effect of verb bias, using cumulative self-paced reading with on-line grammaticality decisions. Presumably, performing grammaticality decisions will slow the overall reading times, as well as accentuate any difficulties. On the NP, reduced sentences were somewhat slower than marked ones, with possibly a larger reduction effect for the SC-biased verbs. On the first word of the disambiguation region, which was an auxiliary verb, there was a small reduction effect for the SC-biased verbs and a very large one for the NP-biased verbs.

Holmes et al. (1989) also used continual grammaticality decision with cumulative displays and compared NP- and SC-biased verbs. However, they added the factor of whether the NP following the verb was a good or bad object. On the NP, the reduced conditions with bad objects tended to be the slowest. For NP-biased verbs, the marked condition with a good object was somewhat faster than the marked condition with a bad object, which is interesting, as one might expect object plausibility to play no role in a marked condition. On the first word of the disambiguating phrase, the marked conditions were faster than the reduced ones. This reduction effect was especially strong for the NP-biased verbs. Noun semantics had little effect at this point, although it appeared that bad objects may have led to slightly faster reading times in the reduced conditions, although this was not significant.

Holmes et al. (1989) then retested these same sentences, asking subjects to try to remember the sentences, rather than perform grammaticality decisions. This change had a significant effect on the results. For SC-biased verbs there was just a reverse reduction effect at the NP, with slower reading of the NP following *that*. For NP-biased verbs, however, there appeared to be an interaction between reduction and noun semantics. Reduced SCs starting with good objects were faster, as were marked SCs with bad objects. These are the two conditions that are least ambiguous at that point, with one appearing to be an NP complement and the other clearly an SC complement. On the first word of the disambiguation, there were almost no effects for SC-biased verbs. For NP-biased verbs there was a clear reduction effect and an apparent interaction between reduction and semantics. Bad objects resulted in faster reading for marked sentences, but slower reading for reduced sentences. It is not clear why this should be the case.

In a third experiment, Holmes et al. (1989) removed the NP semantics condition but examined the effect of lengthening the NP by adding a PP. They also made the change to non-cumulative presentation and question answering, rather than memorization. On the NP there were no large effects, although reading times tended to be shorter for SC-biased verbs. At disambiguation, all of the SC-biased conditions had similar times except for the long, reduced condition, which was slower. For NP-biased verbs, the marked conditions were the fastest, with the reduced sentences having long NPs *faster* than the reduced sentences with short NPs. This latter finding seems to go against the interpretation proposed by the authors, "that when the noun phrase continues for too long, the hypothesis that it is the subject of a new clause is replaced by the direct-object hypothesis." (p. 682)

Ferreira and Henderson (1990), not shown in Table 9.1, used eye-tracking to study the SC ambiguity and found significantly different results than did some of the other studies. Across all of the regions, there was a reduction effect, but little effect of verb bias. In the disambiguation and post-disambiguation regions, the SC-biased verbs actually resulted in slower reading times. A second experiment tested the same sentences using a word-by-word self-paced display. Again there were reduction effects across all regions. Verb-bias effects were small, but numerically the NP-biased conditions were faster on the noun region than the SC-biased conditions, but slower at disambiguation and following. In a final experiment, cumulative displays were used. This resulted in weaker effects, with only a marginal reduction effect at disambiguation. Subjects read faster initially with the cumulative display, and then spent quite a long time at the end of the sentence. The authors suggest that cumulative displays encourage subjects to wait until the entire sentence is visible to process it.

As discussed in Section 3.4, the Ferreira and Henderson (1990) experiments have been criticized on several grounds. Some of the sentences were awkward or implausible, there was a low proportion of filler items, NPs consisted of just one word, comprehension questions were asked sporadically and were not particularly difficult, and, most importantly, the verb bias manipulation was quite weak. In their second experiment, Trueswell et al. (1993) tested self-paced reading using more strongly biased verbs, NPs that consisted of two words and that were good objects of the NP-biased verbs, and more distractor items. On the first word of the NP, *the*, reading times were fastest for the marked conditions, with a reverse verb-bias effect for the reduced conditions, meaning that the SC-biased verbs were slower. On the noun, the reduced SC-biased case was again slower than the other three conditions. In the disambiguation region, there was a small reduction effect on the first word. But on the second word, there was a large slowdown in the reduced NP-biased condition.

In a third experiment, Trueswell et al. (1993) used eye-tracking, rather than self-paced reading. On the NP there was no reduction effect, but NP-biased verbs were nonsignificantly faster than SC-biased verbs. They attribute this lack of effect to preview of later material that is possible when subjects read *that* in marked sentences. In the disambiguation region, there was both a reduction effect and a verb bias effect. The reduced, NP-biased condition was considerably slower than the other conditions.

Trueswell et al. (1993) showed that, in both the second and third experiments, the strength of the reduction effect for SC-biased verbs was correlated with the relative frequency with which the verb occurs with marked and reduced SCs, known as the verb's *that*-preference. If a verb is usually followed by *that*, reduction will tend to have a greater effect.

Garnsey et al. (1997) distinguished verb SC-preference from object plausibility. Using eye-tracking, they compared SC-biased, NP-biased, and equally biased (EQ-biased) verbs and manipulated whether the NP following the verb was a good or bad object. On the ambiguous verb, SC-biased verbs were read faster (first-pass) than the other verb types. On the ambiguous noun phrase there was a reduction effect—NPs were read faster following *that*. One explanation for this is that it is due to parafoveal preview possible when reading *that*. Interestingly, the largest reduction effect was for NP-biased verbs with bad objects and the smallest was for NP-biased verbs with good objects. One might expect the opposite, that reduction effects would be larger for good objects, because a reduced good object with an NP-biased verb should be treated as an object, while a marked one should not. The fact that this is not the case is somewhat perplexing.

In the disambiguation region, the slowest conditions were the reduced NP-biased cases. There was a large reduction effect for NP-biased verbs, but almost no effect for SC-biased verbs. For EQ-biased verbs, the reduction effect was moderate for good objects, but reversed for bad objects. It is not clear why there should be a significant reverse reduction effect for EQ-biased verbs and not for SC-biased. There was a plausibility effect for NP-biased verbs, and a non-significant reverse plausibility effect for SC-biased verbs—bad objects resulted in slower reading at disambiguation. If significant, this too would be hard to explain. Finally, in contrast with the findings of Trueswell et al. (1993), there was no significant correlation between reading times at the NP or disambiguation point and the *that*-preference or frequency of the SC-biased verbs.

The second experiment conducted by Garnsey et al. (1997) measured self-paced reading using the same items. On the NP, there was again a reduction effect. However, in this case the reduction effect cannot be attributed to parafoveal preview in the marked condition. Another explanation is that subjects were sometimes or partially treating the reduced cases as SCs, resulting in a slowdown due to preparation or semantic restructuring. In the marked case, these processes could have been partially accomplished on *that*. However, this explanation should probably lead to the prediction that reduced SC-biased verbs would be the slowest. Rather, it was the reduced NP-biased verbs that were the slowest, which is again perplexing. There did not seem to be a strong effect of noun semantics in reading times of the NP.

In the disambiguation region, the slowest cases, once again, were the reduced NP-biased verbs, especially with good objects. The effects of reduction and semantics were weak or non-existent among the other conditions. The results of this second experiment agree quite well with those of the first. One notable difference is that the odd reverse-ambiguity effect with EQ-biased verbs and good objects did not replicate. Interestingly, the garden-path effects in these two experiments were quite small. In the second experiment, the reduced NP-biased conditions were only about 6% slower, on average, than the other conditions. Thus, in contrast with other ambiguities, readers do not appear to be strongly garden-pathed by the SC ambiguity, although it may be possible to engineer a strong effect with especially difficult items.

Although both experiments of Garnsey et al. (1997) found little or no effect of plausibility on the sentences with SC-biased verbs, Pickering et al. (2000) did find such an effect in an eye-tracking study. Their items contained a post-NP phrase, such as *sometime* or *one day*, which extended the ambiguity. Only SC-biased verbs were used with reduced SCs. On the NP, the implausible objects were non-significantly slower in first-pass reading times, but there were more regressions and they were significantly slower in overall reading time. On the post-NP phrase, the differences were more pronounced. At the disambiguation point, the results were reversed, with a non-significant slowdown in first-pass reading for good objects, but larger effects in other measures. A second experiment placed the experimental items in a context. Similar results were found on the NP and post-NP, but there were no significant effects on the SC verb.

9.1.1 Empirical results summary

Because it deals with three factors—reduction, verb bias, and object plausibility—the experimental work on the SC ambiguity leads to somewhat more complex patterns of results than we faced with the MV/RR ambiguity. In the ambiguous region, Holmes (1987), Trueswell et al. (1993) Expt. 2, and Garnsey et al. (1997), both experiments, found apparent reduction effects. However, Holmes et al. (1987), Holmes et al. (1989) Expt. 2, and Trueswell et al. (1993), Expt. 3, found no effect of reduction or a reverse effect during the ambiguity. The effects of verb bias during the ambiguity are rather inconsistent, with most experiments showing no clear main effect. Trueswell et al. (1993), Expt. 2, found the reduced SC-biased sentences harder than the reduced NP-biased, and in Expt. 2 they found both types of SC-biased sentences harder, with no effect of reduction. There seemed to be no clear pattern of overall effects of verb-bias or interactions with reduction and NP plausibility in the Garnsey et al. (1997) experiments, except possibly for slower reading of bad objects in reduced SCs following NP-biased verbs versus SC-biased verbs. The effects of NP plausibility, or semantics, were also weak and inconsistent. Pickering et al. (2000) appeared to find slower reading of bad objects in reduced SCs following SC-biased verbs. However, Garnsey et al. (1997) and Holmes et al. (1989), Expt. 2, found little or no difference between these conditions.

The findings are somewhat more consistent in the disambiguation regions. All studies that included the appropriate conditions found that reduced NP-biased verbs resulted in slower reading at disambiguation than marked NP-biased verbs. The reduction effect for SC-biased verbs is much weaker, with about a third of the experiments finding slightly faster marked conditions, a third finding slightly faster reduced conditions, and the others finding no effects. All of the experiments find reduced NP-biased verbs to result in slower disambiguation than reduced SC-biased verbs. The difference between the marked SC-biased and NP-biased conditions is again much weaker, although there are a few more studies showing slower reading in marked NP-biased conditions than there are studies showing slower reading in marked SC-biased conditions.

Finally, we turn to effects of NP object plausibility at the disambiguation point. Holmes et al. (1989), Expt. 1, found slower reading after good objects in reduced, but not in marked, conditions. In the second experiment, good objects led to *faster* reading times in reduced conditions but slower reading times in marked conditions. In both experiments, Garnsey et al. (1997) found good objects to result in slower reading at disambiguation for NP-biased verbs, but slightly faster reading for SC-biased verbs. The effects of plausibility were not as strong for SC-biased verbs as they were for NP-biased verbs. Thus, the effects of object plausibility at disambiguation appear to be relatively weak and they interact with other factors in inconsistent ways.

9.2 Experiment 7

In order to study the ability of the CSCP model to resolve the SC ambiguity, an experiment was conducted manipulating three factors: reduction, verb bias, and NP plausibility. All sentences were of similar form to (56). The subject always consisted of the definite article and a noun, either singular or plural. This was followed by the main verb and then, in marked conditions, by the complementizer *that*. The SC was always a passive sentence in the simple past tense with a verb-modifying PP. The subject of the SC consisted of an article (not always definite) and a noun, singular or plural. The SCs for each subject noun were not assembled piecewise, but were extracted from a large set of Penglish sentences using TGREP2. This was to ensure that the semantics of the SCs was reasonable, although it does not entirely eliminate bias between conditions that use different sets of nouns for the SC subjects. Passive SCs were used because more nouns can be used plausibly as the subject of a passive than of an active verb, and the first word of the disambiguation point is either *was* or *were* and is thus consistent across conditions. Sentences of this form involve four propositions.

(56) The mother found [that] the food was put on the table.

The first manipulated factor is whether the SC is marked or reduced. The next is whether the main verb is SC-biased or NP-biased, based on the relative frequency with which the verb is followed by an SC or an NP complement. Table 9.2 shows the five SC-biased and the six NP-biased verbs used in this study. The SC-biased verbs range from an SC-preference of 89.8% for *believed* to 47% for *showed*, which is therefore not strictly SC-biased. The NP-biased verbs range in SC-preference from 30.3% for *found* to 8.6% for *read*. The SC-biased verbs have somewhat lower mean log overall frequency (8.85 vs. 9.29) and lower NP-complement frequency (7.73 vs. 9.08), but higher SC

Verb	NP	SC	% SC	Reduced SC	% RSC
believed	390	3,421	89.8%	2,248	65.7%
forgot	562	878	61.0%	340	38.7%
knew	8,431	11,874	58.5%	2,530	21.3%
felt	4,117	5,575	57.5%	3,142	56.4%
showed	8,094	7,183	47.0%	1,555	21.6%
found	17,428	7,583	30.3%	2,253	29.7%
saw	20,089	6,629	24.8%	852	12.9%
wrote	6,582	1,962	23.0%	526	26.8%
heard	5,473	783	12.5%	292	37.3%
told	12,241	1,282	9.5%	559	43.6%
read	2,987	280	8.6%	129	46.1%

Table 9.2: Frequency of verb use with NP and sentential complements per 1 million Penglish sentences and the relative frequency of SCs. Also shown is the frequency and percentage of reduced SCs.

frequency (8.36 vs. 7.46).

Several Penglish verbs that permit SCs could not be used in this study. *Realized*, *hoped*, and *wished* do not allow direct objects, *questioned* and *asked* use only marked SCs, and *said*, *thought*, and *guessed* were eliminated because the set of direct objects that they allow is so limited. For example, the only noun complements of *said* are *something* and an *answer*.

The third factor manipulated is whether the NP following the main verb, which is actually the subject of the SC in all cases, is potentially a good or bad object of the verb. For each verb, eight good objects and eight bad objects were chosen. The good objects were the eight nouns most commonly appearing as objects of the verb in a Penglish corpus. The bad objects were chosen from the set of nouns that could serve as objects of at least one verb and as subjects of an SC, but are never used as objects of the verb in question. If eight such nouns could not be found, then the objects were drawn from the nouns that could serve as objects of the verb, but do so least frequently. It should be mentioned that there is a complicating factor overlooked in the design of this experiment. The verbs *showed*, *wrote*, *told*, and *read* can act as either transitives or ditransitives. Some of the bad objects for these verbs could have been plausible indirect objects, if not plausible direct objects. This introduces an additional temporary ambiguity that may have affected the comprehension and reading time performance. Nevertheless, due to the limited pool of available SC/NP verbs, the ditransitives were included.

Fifty sentences were produced for each of the four conditions, crossing reduction and NP plausibility, for each verb. All three networks were tested on these sentences and their results were averaged.

9.2.1 Comprehension results

Although there are few available empirical results for comparison, we will begin with the comprehension performance of the model. Figure 9.1 shows the average question-answering error rates across the eight conditions, and Figure 9.2 shows the main effects of the three factors, as well as interactions between pairs of factors. Both the strict and multiple-choice error rates are shown. Note, first of all, that the error rates are quite low in comparison to those for the MV/RR ambiguity. The strict-criterion error rate averages about 13.5% while the multiple-choice rate averages about 2%. Apparently, the model has little difficulty with sentential complements, reduced or otherwise.

All three factors show main effects in the expected directions. Marked sentences are easier than reduced ones,[1] SC-biased verbs are considerably easier than NP-biased verbs,[2] and bad objects are easier than good objects.[3] As shown in the upper right of Figure 9.2, there is a clear interaction between verb bias and reduction.[4] Reduction has

[1] Strict: $F(1,548)=95.5$, $p<0.001$; Multiple-Choice: $F(1,548)=21.3$, $p<0.001$
[2] Strict: $F(1,548)=19.5$, $p<0.001$; Multiple-Choice: $F(1,548)=41.0$, $p<0.001$
[3] Strict: $F(1,548)=11.7$, $p<0.001$; Multiple Choice: $F(1,548)=5.33$, $p=0.021$
[4] Strict: $F(1,548)=51.3$, $p<0.001$; Multiple-Choice: $F(1,548)=7.91$, $p=0.005$

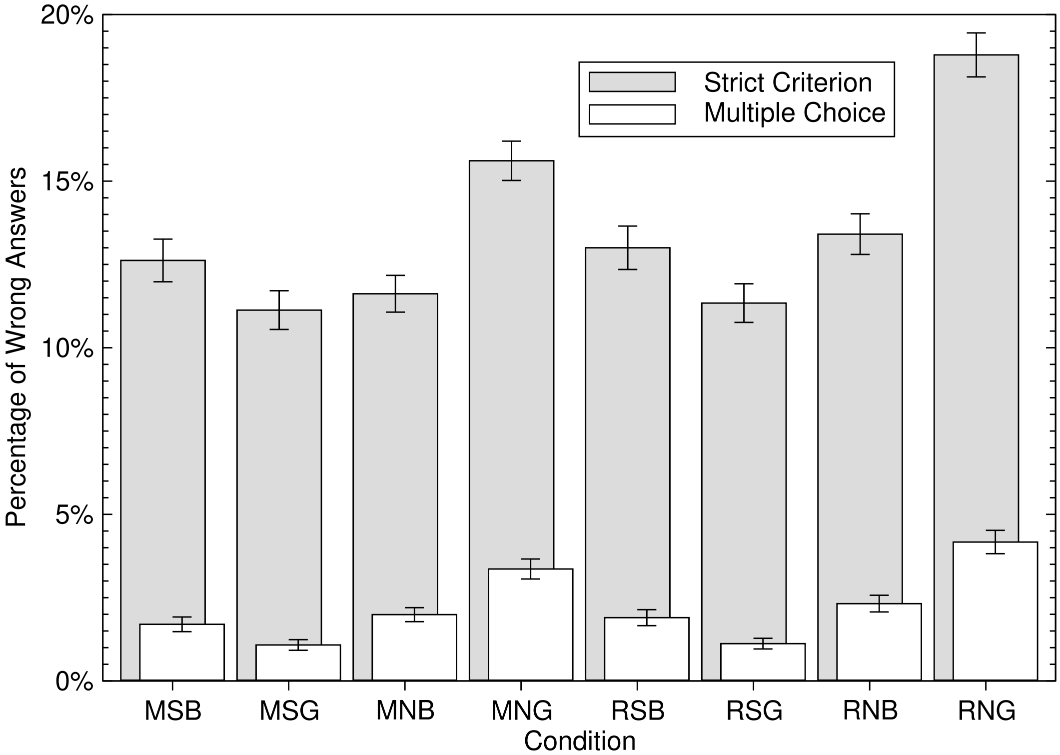


Figure 9.1: Comprehension error across all conditions in Experiment 7.

virtually no effect on SC-biased verbs, but a large effect on NP-biased verbs. Accordingly, verb bias has a moderate effect on marked SCs but a strong effect on reduced SCs.

The interaction between reduction and NP plausibility is much weaker and is only mildly significant for the strict criterion and non-significant using multiple-choice.[5] There is a hint that the reduction effect is larger for good objects, but this is primarily a case of two main effects. The same is not true, however, of the interaction between verb bias and NP plausibility.[6] For NP-biased verbs, the good object condition is much harder than the bad object condition. But for SC-biased verbs the good object condition is actually *easier* than the bad object condition. The reason for this may be that a noun that is a good object also makes a better subject of the passive sentential complement. Therefore, if this bias were removed by mixing active and passive SCs, the plausibility effect on the NP-biased verbs might become much stronger. It should be noted that this reverse plausibility effect for the SC-biased verbs did not appear in an earlier version of this experiment which used a different criterion for choosing bad objects. Thus, the effect may be specific to these items.

9.2.2 Reading time results

Reading time results for the sentential complement ambiguity are displayed in Figures 9.3–9.6. SRTs for the marked conditions are shown in Figure 9.3. There are no effects of verb bias on the verb itself. However, *that* is read more quickly following SC-biased verbs. There are no large effects on the article, although it may be slightly slower for NP-biased verbs. On the ambiguous NP there is an interesting interaction. The fastest condition is for the NP-biased verbs with good objects and the slowest is for NP-biased verbs with bad objects. The effects of object plausibility are weaker for SC-biased verbs. Similar, although not identical, results were found in the two Garnsey et al. (1997) experiments. Their first experiment found, among the marked conditions, that the MNG condition was the slowest during the ambiguity, with the MNB condition fastest and the MS conditions in between and of similar rate. Their second experiment also found the MNB condition to be the fastest, but there was little difference in the other conditions.

[5] Strict: $F(1,548)=5.12$, $p=0.024$; Multiple-Choice: $F(1,548)=1.46$, $p=0.23$

[6] Strict: $F(1,548)=33.4$, $p<0.001$; Multiple-Choice: $F(1,548)=22.5$, $p<0.001$

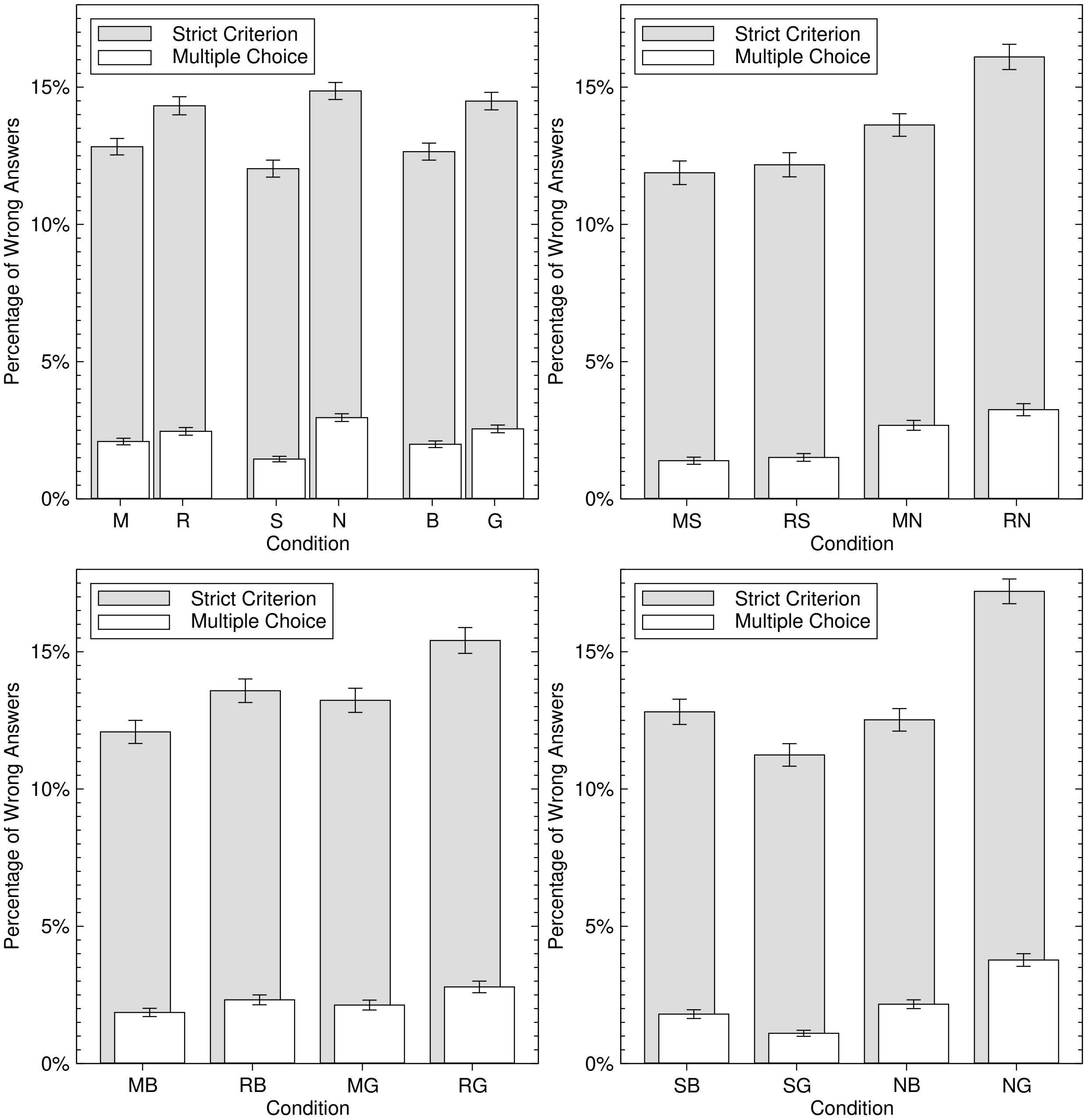


Figure 9.2: Main effects of the three factors in Experiment 7 and interactions between each pair of factors.

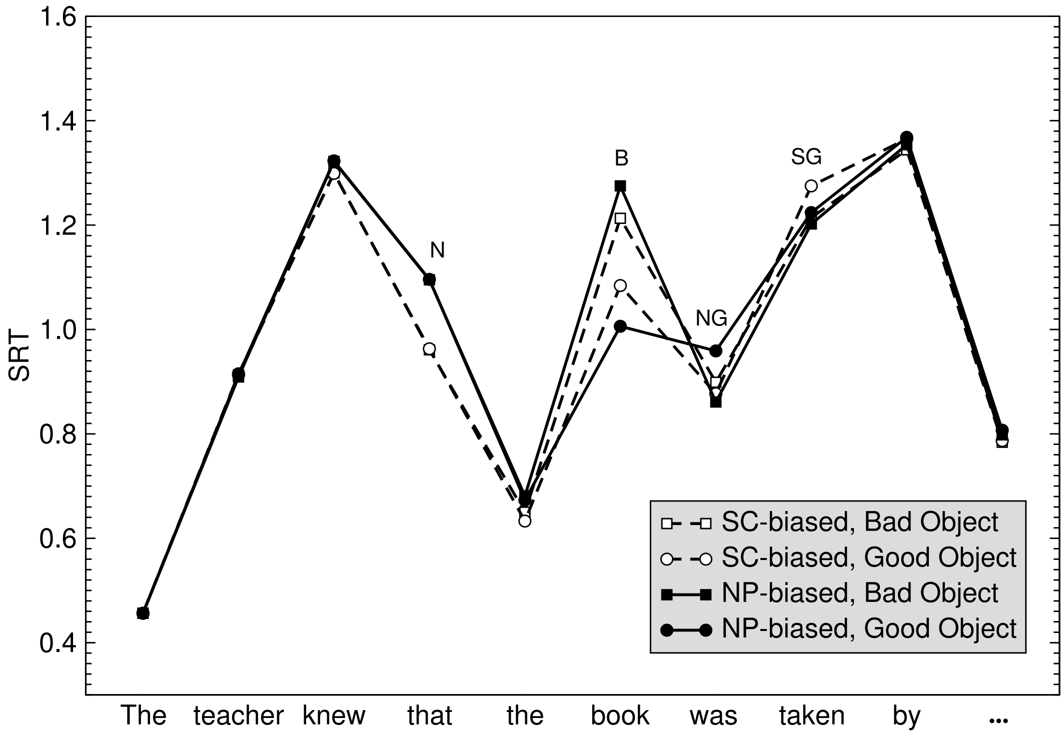


Figure 9.3: Reading times for marked SC conditions in Experiment 7.

As expected, at the disambiguation point the reading times for the good and bad objects with NP-biased verbs crossed, with the MNG condition being the slowest and little difference in the other conditions. This exactly mirrors the findings of Garnsey et al. (1997), Expt. 2. Their first experiment found the MSB condition to be slightly slower than the MNG.

Figure 9.4 shows the corresponding results for the reduced SCs. The article is read slightly slower following SC-biased verbs. Presumably this is because *that* was, to some extent, expected. At the ambiguous NP, the bad objects are read slower than the good objects. The fastest condition is for the good objects with NP-biased verbs, as is to be expected. Garnsey et al. (1997), Expt. 1, found similar results, with the RNB condition slowest and the RNG condition fastest. However, they did not find the RSG condition to be faster than RSB. Garnsey et al. (1997), Expt. 2, did agree with the current findings on that point, but, surprisingly, found the RNG condition to be slowest.

At the disambiguation point there is a strong effect of verb bias, but little effect of NP plausibility, although the slowest condition appears to be the RNG. Both of the Garnsey et al. (1997) experiments also found the RNG condition to be slowest at disambiguation and the RNB condition to be the next slowest. However, those experiments did find a reverse plausibility effect for the SC-biased verbs, while the model shows no such effect. In both the marked and reduced sentences, the SG condition was slower than the other conditions post-disambiguation. It is not clear why this is the case, but it may be related to the fact that the SG conditions were also comprehended best. It may be that the model is doing some deeper processing of the SC in this case. If replicable, it may be worth looking into this further.

Figure 9.5 compares the marked and reduced conditions, with verb bias as a factor and collapsing across NP plausibility. On the article, there is a reduction effect and, for the reduced conditions, the SC-biased verbs are slower. Presumably this is because an article is to be expected following *that* and an article is more likely immediately after an NP-biased verb. On the ambiguous noun there is a small reverse reduction effect. Across the NP, however, the model predicts that reduced sentences should be slower than marked and that the RS condition, in particular, will be the slowest, with a smaller effect of verb bias on the marked condition. This agrees perfectly with the Trueswell et al. (1993), Expt. 2, findings and is consistent with Holmes (1987) and Garnsey et al. (1997), Expt. 1. However, the empirical results are inconsistent on this point and the model does not agree with all of the studies. Holmes et al. (1987) found the RN condition to be the slowest on the NP, Trueswell et al. (1993), Expt. 3, found the MS condition to be harder than the RN condition, Garnsey et al. (1997), Expt. 2, found the RN condition to be easier than RS, and

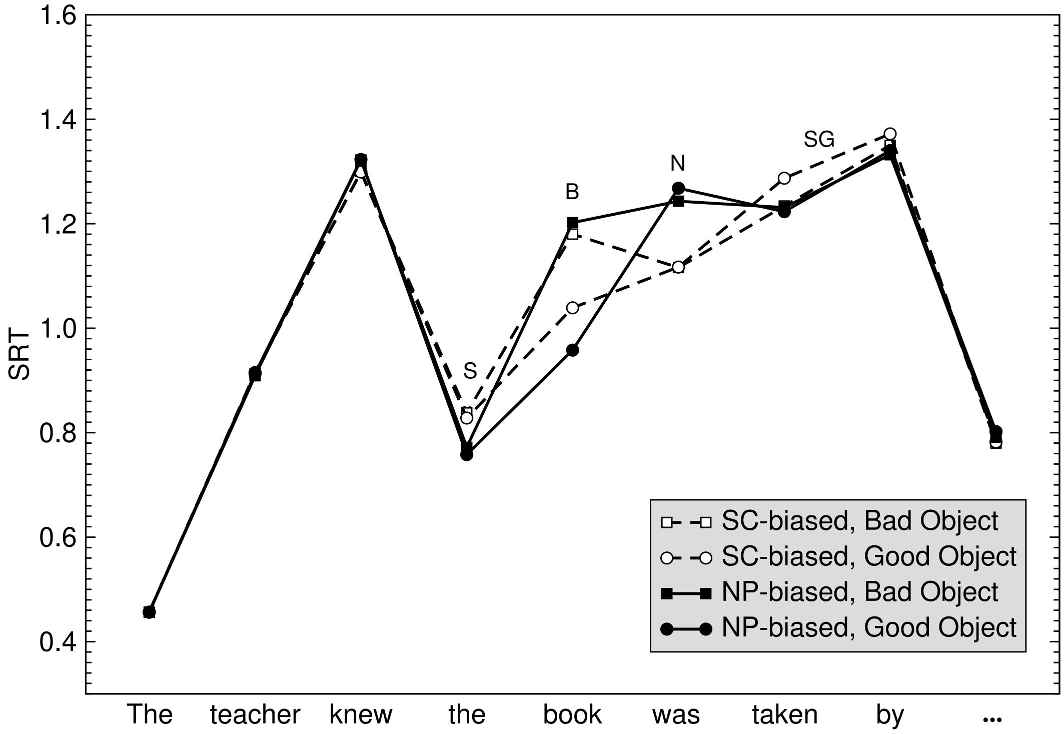


Figure 9.4: Reading times for reduced SC conditions in Experiment 7.

Holmes et al. (1989) obtained quite different results than the other experiments. But the model does seem to be consistent with the overall trend.

At the disambiguation point, there is little difference between the marked conditions but a large effect of reduction, with the RN condition by far the slowest. This matches the results of Holmes (1987), Holmes et al. (1989), Expt. 1, and Trueswell et al. (1993), Expts. 2 and 3. The other experiments agree that the RN, with bad or good objects, should be the hardest conditions, but they differ in the relative difficulty of the RS and the marked conditions.

Figure 9.6 shows the reduction and noun plausibility conditions, collapsing across verb bias. On reading the noun, there is a small reverse reduction effect and a large reverse plausibility effect. The good objects are read faster than the bad objects. The reverse reduction effect on the noun is smaller than the reduction effect on the article, so the net result would be a reduction effect across the NP. The fact that the RB condition is the slowest is consistent with Holmes et al. (1989), Expt. 1, and Pickering et al. (2000). Holmes et al. (1989), Expt. 2, and the Garnsey et al. (1997) experiments seem to show no consistent effect of noun plausibility in reading times of the NP.

At disambiguation there is a large reduction effect. However, the overall effect of noun plausibility on reading time is very small, with the good objects only slightly slower than the bad objects. As seen in the previous figures, the plausibility effect at disambiguation is almost entirely due to the NP-biased verbs. The plausibility effect was also seen at disambiguation in Holmes et al. (1989), Expt. 1, Pickering et al. (2000), and for the most difficult RN conditions in the two Garnsey et al. (1997) experiments. These experiments produce weak or inconsistent results in the other conditions. One exception to these findings is the Holmes et al. (1989), Expt. 2, which produced a reverse plausibility effect for the RN conditions.

9.2.3 Experiment 7b: Lengthening the NP

Before summarizing, let us turn briefly to one last experimental manipulation: that of lengthening the NP. A natural prediction might be that lengthening the ambiguous NP with a post-modifier would strengthen the noun-phrase interpretation and lead to longer reading times at disambiguation in all conditions. An alternative prediction is that a longer NP will lead to greater commitment to whichever interpretation is preferred at the time of the NP. This might mean

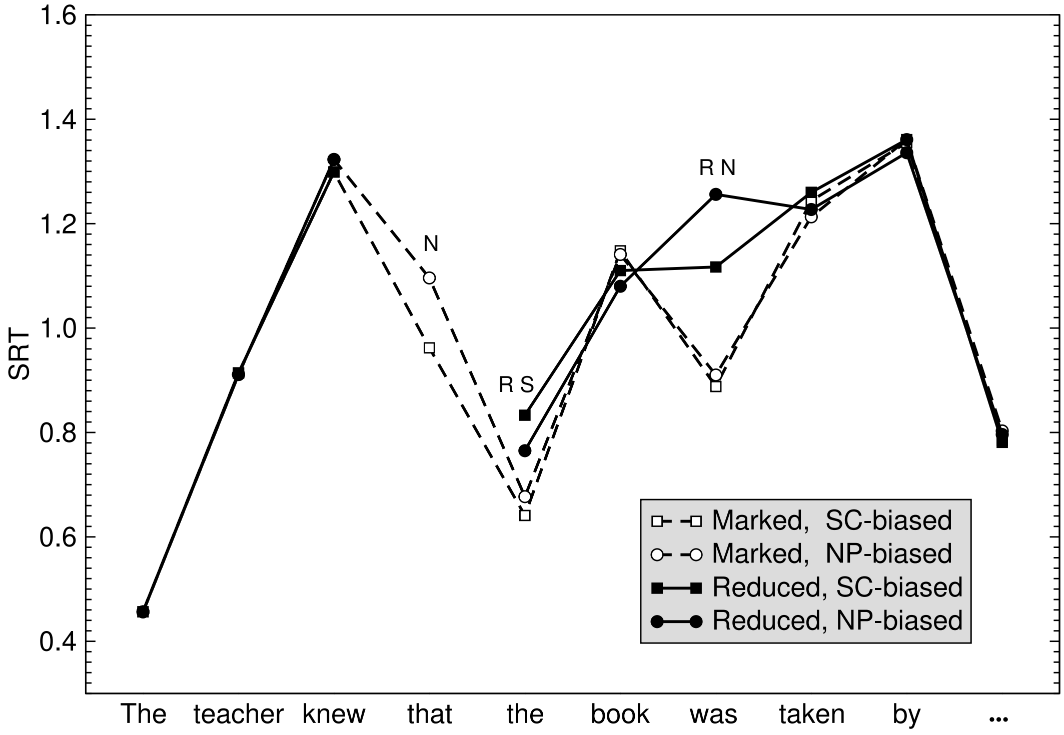


Figure 9.5: Reading times in Experiment 7, collapsing across NP plausibility.

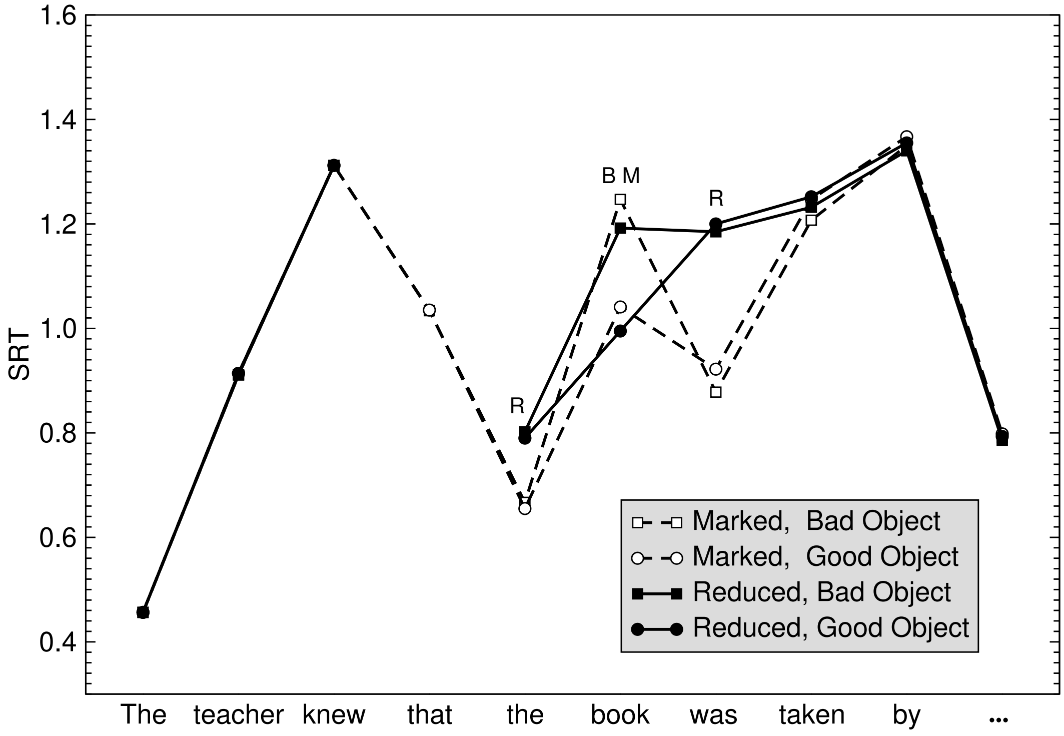


Figure 9.6: Reading times in Experiment 7, collapsing across verb bias.

that the longer NP leads to slower reading times for NP-biased verbs with a reduced SC, but longer reading times for SC-biased verbs, or perhaps for all verbs in marked conditions.

However, as discussed earlier, when Holmes et al. (1989) tested the effect of extending the NP with a PP, the results did not accord with either of these predictions. There was little effect of NP length in the marked conditions. In the reduced conditions, the longer NPs lead to *slower* reading at disambiguation for the SC-biased verbs but *faster* reading at disambiguation for the NP-biased verbs. This seems to be exactly opposite the findings one would expect if the effect of the extended NP was to induce greater commitment to the preferred hypothesis. Although more empirical studies are called for to confirm these findings, it seemed worthwhile to test the model's prediction of the effect of extending the NP.

The items in this portion of the experiment were generated in identical fashion to those used earlier, with the exception that each ambiguous NP was modified by a three-word PP. For each noun, the modifying PPs were drawn from a large corpus of Penglish, so the semantics of the relationships should be fairly natural. The length factor was crossed with the three factors previously used, resulting in sixteen total conditions. The symbol `L` will be used to indicate conditions with long NPs and the symbol `H` will be used to indicate the conditions with short NPs, which are those we have already discussed.

In terms of comprehension, the results are not particularly interesting. In order to make a more balanced comparison between the short and long conditions, answers to questions about the added PPs were excluded. The additional semantic and syntactic burden of the PP is expected to decrease performance in answering other questions about the sentence. Indeed, the PP has a significant impact, raising the strict error rate from 13.6% to 26.7% and the multiple-choice error rate from 2.3% to 5.9%. Nevertheless, the error rate for the long NP sentences is still quite small for the model in comparison to its performance on other 5-proposition sentences.

The interaction between NP length and reduction is significant using either error measure. The interaction between length and verb bias is significant only according to the strict error measure and the interaction between length and NP plausibility is significant only using multiple-choice. All interactions are in the direction of a greater effect of NP length on the harder of the two conditions. Thus, lengthening the NP has a stronger effect on reduced SCs than it does on marked SCs. This is consistent with the hypothesis that lengthening the NP increases the strength of the NP interpretation as well as with the hypothesis that it increases the strength of the preferred hypothesis, whichever that may be. More interesting results can be found in the analysis of reading times.

Because the short and long sentences are similar, though not identical, up to the NP, there are no interesting reading-time analyses to be done before the end of the NP. Therefore, we will focus on reading times at the disambiguation point. Figure 9.7 shows the combined SRTs on the first two words of the disambiguation (*was put*) across the 16 conditions. Lengthening the NP leads to slightly longer reading times at disambiguation for the marked SCs, as might be expected. However, the longer NPs result in considerably *shorter* reading times in the reduced conditions.

One possible explanation for this behavior in the model is the following. Rather than leading to greater commitment to one hypothesis over another, the effect of lengthening the NP may be to reduce the model's commitment to either hypothesis. The result might be considered a regression to the mean. In the marked case, in which the model should mainly be expecting an SC, the longer NP weakens this expectation and results in greater surprise at the disambiguation point. In a reduced sentence, the model may be more committed to the NP hypothesis and is thus treating the NP as a direct object. But the PP weakens this interpretation, leading the model to be more open to the possibility that the NP is a subject. The result is a reduced garden-path effect at disambiguation. It is likely that the effect of the longer NP will be stronger for reduced sentences where the model may be less committed to either hypothesis.

This "regression to the mean" hypothesis also provides an explanation for the results of the Holmes et al. (1989) experiment, in which longer NPs led to slower reading at disambiguation for SC-biased verbs but faster reading for NP-biased verbs. In the SC-biased case, subjects may have preferred the correct SC interpretation, which was weakened by the PP. In the NP-biased case, subjects may have preferred the incorrect NP reading. When this was weakened by the PP, there was less surprise at disambiguation. Note that faster reading times do not necessarily mean the subjects understood the sentences better. That they were, by hypothesis, less committed to the wrong interpretation does not mean they are more committed to the correct interpretation. It may be that, if comprehension were carefully measured, subjects, like the model, would experience greater difficulty understanding the sentences with long NPs than those with short NPs.

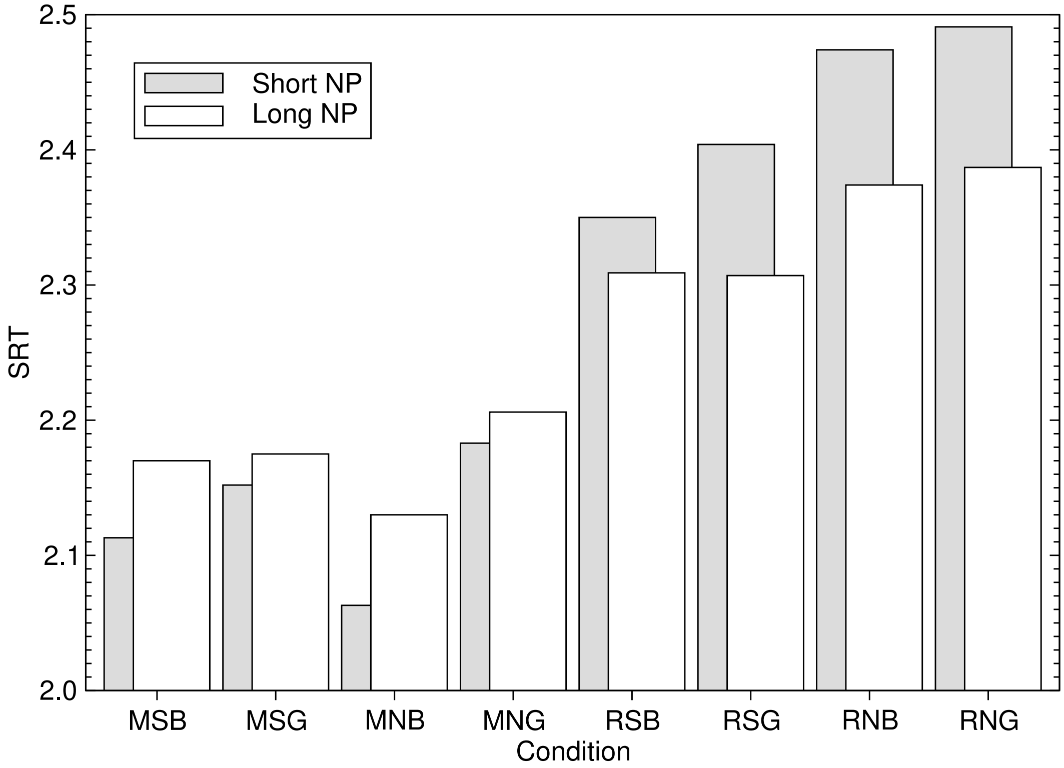


Figure 9.7: Reading time on the first two words of the disambiguating VP for all sixteen conditions.

9.3 Summary

This section summarizes the model's behavior on the sentential complement ambiguity experiments. Comprehension error rates are quite low across the board for sentences of this type. This may be a combination of the facts that sentential complements are quite frequent in Penglish, they are often reduced, and the word order in an SC sentence is identical to standard matrix clause order. There were main effects of the four manipulated factors: marked sentences are easier than reduced, SC-biased verbs are easier than NP-biased verbs, bad objects are easier than good objects, and short NPs are easier than long NPs. There were also strong pairwise interactions between reduction and verb bias, verb bias and NP plausibility, and reduction and NP length, and weak interactions between reduction and NP plausibility, verb bias and NP length, and NP plausibility and NP length. There were also some significant higher-order interactions.

In terms of reading times, there were interesting effects at both the ambiguous NP and in the disambiguating region. On the article of the NP there is a reduction effect (the reduced condition is slower than the marked), but on the noun itself there is a weaker reverse reduction effect. The net result is a moderate reduction effect across the NP. The majority of experimental studies also found reduction effects in the ambiguous region. On the article there is also a reverse verb bias effect (SC-biased is slower than NP-biased), which is stronger in the reduced case. A similar effect was found across the ambiguous region in Trueswell et al. (1993), Expt. 3, Holmes et al. (1989), Expt. 3, and in reduced conditions in Trueswell et al. (1993), Expt. 2. Most other studies, however, found inconsistent effects of verb bias in the ambiguous region in different conditions.

On the ambiguous noun, there is a strong reverse effect of NP plausibility and an interaction between verb bias and NP plausibility, such that the slowest condition is the NP-biased with a bad object and the fastest is the NP-biased with a good object. For SC-biased verbs, the bad objects are still slower than the good objects, but the difference is smaller. Effects of plausibility on reading times in the ambiguous region are unclear in most experimental studies reviewed here, although Pickering et al. (2000) did also find a reverse plausibility effect, meaning that the good objects are read faster than the bad objects.

In the disambiguation region there is a large effect of reduction which interacts with verb bias. The slowest condition is reduced and NP-biased followed by reduced and SC-biased. There is little or no verb bias effect in the marked conditions. The reduction and verb bias effects, as well as the greater verb bias effect in reduced conditions, appear to be supported by most of the empirical studies reviewed here.

The model experiences a relatively weak effect of NP plausibility at disambiguation. However, it seems to interact with verb bias such that there is a stronger plausibility effect for the NP-biased verbs. The empirical data generally supports the finding that good objects result in slower reading at disambiguation, and that the effect is stronger, or may only exist, for the reduced NP-biased conditions.

The effect of lengthening the NP on reading times at disambiguation depends on whether the SC is reduced. Long NPs result in slower reading in marked conditions but faster reading in reduced conditions. A possible explanation proposed for this is that lengthening the NP results in noise or confusion that leads to a weakening of commitment to the preferred interpretation, which is reflected in a regression to the mean in terms of reading times.

Chapter 10

The Subordinate Clause Ambiguity

The last temporarily ambiguous structure to be examined is the subordinate clause, or NP/0, ambiguity. This occurs when a subordinate clause with an intransitive verb that is optionally transitive precedes a main clause, with no prosodic or punctuated boundary marking. An example of this type of sentence is shown in (57a). It should be noted that this type of ambiguity is somewhat unnatural in written English because it is standard to place a comma between the two clauses.

(57a) After the private saluted the sergeant decided to end the military drill.

(57b) After the private saluted the sergeant the captain decided to end the military drill.

Three factors affecting the difficulty of this ambiguity have been investigated. The first is whether the subordinate-clause verb (*saluted*) is strongly transitive, weakly transitive, or unambiguously intransitive. It is expected that the ambiguity will be more difficult if the verb frequently occurs in transitive form. The next factor, as in the SC ambiguity, is whether the NP following the verb is a plausible object of the transitive form of the verb. The good objects are expected to cause more difficulty than the bad objects. Finally, the length of the ambiguous NP may also play a role.

10.1 Empirical results

Frazier and Rayner (1982) used eye-tracking to investigate this ambiguity, comparing "early-closure" sentences, such as (57a), with "late-closure" sentences, such as (57b), in which the NP actually is the object of the verb. Also included were short and long NP conditions. Although they found shorter reading times at disambiguation for the late-closure sentences, the comparison is not particularly reliable as the two conditions did not share similar disambiguation regions. The authors did also find that reading times were slower at disambiguation following long NPs. Table 10.1 summarizes these and the other experimental results.

Mitchell and Holmes (1985) conducted several phrase-by-phrase reading experiments which, among other conditions, investigated the effect of verb bias on the NP/0 ambiguity. Reading times were reported only on the first two words of the disambiguating verb phrase (*decided to*). As expected, transitive-biased verbs resulted in longer reading times at the point of disambiguation. A second experiment presented the sentences as a whole and measured overall reading time. Again, the transitive-biased verbs resulted in longer reading times. When a comma was inserted to disambiguate the sentences, the reading time difference at disambiguation was smaller than in the first experiment, but it was still apparent, numerically. The authors did not report significance tests for this condition alone.

Mitchell (1987) conducted further phrase-by-phrase reading experiments using intransitive and optionally transitive verbs. Displays were partitioned into two segments, with the division falling either before or after the ambiguous NP. When the division fell before the NP, at the clause boundary, the intransitive conditions were slightly slower on the first half and slightly faster on the second half, but the difference was not significant. When the displays were segmented after the NP, these differences were much stronger and were significant. The intransitive verbs, in comparison to the transitive ones, resulted in slower reading on the first half of the display, but faster reading on the second half of the display.

Study	Method	Ambiguous Region
FR82	ET	H $\leq$ L
M87	PSP	S $<$ I
PT98-1	ET	G $<$ B
PTC00-3	ET	WG $<$ WB
CSCP Model	SP?	SG $<$ WG $\leq$ SB $<$ WB $<$ I
Study	**Method**	**Disambiguating Region**
FR82	ET	H $<$ L
MH85-1	PSP	W $<$ S
M87	PSP	I $<$ S
PT98-1	ET	B $<$ G
PTC00-3	ET	WB $<$ WG
CSCP Model	SP?	I $<$ WB $<=$ SB $<$ WG $<$ SG

Table 10.1: Summary of reading-time results on the subordinate clause ambiguity. Methods: PSP = phrase-by-phrase self-paced reading, ET = eye-tracking. Conditions: S = strongly transitive, W = weakly transitive, I = intransitive only, B = bad (implausible) object, G = good object, H = short NP, L = long NP.

Pickering and Traxler (1998) conducted two eye-tracking studies of the effect of NP-plausibility on this ambiguity. In the first, the ambiguous verbs were controlled so that the ambiguous nouns following them were either good or bad objects. The nouns were also modified by prepositional phrases. In the noun and post-noun regions, first-pass reading times and the frequency of regressions were higher for implausible objects. In the verb and post-verb regions, the opposite was true, although the effect in first-pass reading times at disambiguation was very small. In a replication, most of these effects reappeared, although first-pass reading times in the post-verb region were slower for plausible objects than for implausible ones and there was no first-pass effect in the post-verb region.

Finally, Pickering et al. (2000), Expt. 3, performed another eye-tracking study of the effect of NP plausibility on the NP/0 ambiguity. The ambiguous nouns were controlled to be either good or bad possible objects of the verbs. The nouns were again followed by an adverbial phrase that did not resolve the ambiguity. All verbs in this study were optionally transitive, but preferred the intransitive form. On the noun and in the post-noun region, the bad object condition was slower than the good object condition. However, this difference was not significant in first-pass reading times, only in right-bounded and regression-path reading times. At the disambiguation point, the results reversed and the good object condition was significantly slower in first-pass and overall reading times.

Experimental measures of reading time on the subordinate clause ambiguity are not as abundant as those for the MV/RR and SC ambiguities, but the general results do appear to be in line with what we would expect. Conditions biased toward the late-closure reading tend to be faster on the ambiguous NP, but slower during disambiguation, than conditions biased toward the early-closure reading. Thus, strong transitive verbs and good objects result in bigger garden-path effects. Longer ambiguous NPs may lead to slower reading both before and after disambiguation, but that is likely to depend on the manner in which the NP was lengthened. Unfortunately, none of the studies reviewed here crossed multiple factors, so we do not know what sort of interactions ought to occur between those factors.

10.2 Experiment 8

This first experiment on the subordinate clause ambiguity used sentences of similar form to (58). Each one contained an intransitive subordinate clause followed by a passive main clause. Subordinate clauses began with *while*, *because*, or *although*. Three classes of subordinate clause verbs were used: strongly transitive, weakly transitive, and intransitive. Considering just intransitive and transitive forms, the strong transitive verbs had a transitive frequency of at least 76% in Penglish. The weak transitive verbs had a transitive frequency under 56%. The intransitives, of course, cannot be used transitively. One exception to this is that *flew* has both transitive and intransitive senses. The subjects of flew were constrained to be nouns (*bird, birds, plane, planes*) that can only serve as intransitive subjects. Some verbs were eliminated because they permit only a very small set of objects in their transitive form, and some were eliminated

Verb	Intrans	Trans	% Trans
asked	720	10,177	93.4%
bought	1,794	16,780	90.3%
threw	415	3,411	89.2%
told	1,633	12,241	88.2%
killed	1,124	7,906	87.5%
gave	1,398	8,788	86.3%
hit	1,008	5,137	83.6%
saw	6,275	20,089	76.2%
forgot	438	562	56.2%
knew	6,788	8,431	55.4%
bit	624	723	53.7%
ate	824	839	50.4%
guessed	215	78	26.6%
believed	1,078	390	26.6%
*flew	3,144	539	14.6%
thought	10,541	1,185	10.1%
went	20,011	0	0%
hoped	1,482	0	0%
barked	977	0	0%

Table 10.2: Frequency of verb use in intransitive and transitive forms per 1 million Penglish sentences. The upper set of verbs is the strong transitives, followed by the weak transitives and the intransitives.

because they require a prepositional phrase in their intransitive form. Table 10.2 lists some frequency information for the three classes of verbs used in the experiment.

(58) Although the teacher saw a book was taken in the school.

For the transitive verbs, two sets of nouns were used for the subject of the main clause (*book*). The good objects were nouns that could plausibly serve as direct objects of the transitive form of the verb. These were chosen from the 8 nouns that most frequently occur as direct objects of the verb, or all nouns that occur as direct objects, if fewer than 8. The bad objects included all nouns that were not among the top 16 most frequent objects. Of course, the intransitive verbs, other than *flew*, never take direct objects, so they only have a bad object condition. The subjects of the ambiguous verbs were selected from the set of nouns that can serve as the subject of the verb in its intransitive form. For the weak and strong transitive verbs, most of these nouns could also serve as subjects of the transitive form.

The verb sets did not differ greatly in terms of frequency of their intransitive forms. The average log frequency of the verbs in intransitive form was 7.2 for the strong transitives, 6.7 for the weak transitives, and 8.3 for the intransitives. The average log transitive frequencies were 9.1 for the strong transitives and 6.5 for the weak transitives. The combined intransitive and transitive frequencies were 9.3 for the strong, 8.3 for intransitive, and 7.4 for the weak transitive. Thus, the strong transitives were the most common and the weak transitives were the least common.

The main clauses were all in passive past tense form and started with *was* or *were*. Each clause also included a by-phrase or a prepositional phrase modifying the verb. The main clauses were extracted from a corpus of Penglish to ensure reasonably plausible semantics. For each combination of a verb and, for the transitives, one of the two object plausibility conditions, 50 sentences were generated, for a total of 1,650 items. All three networks were tested on the full set of items and their results averaged.

10.2.1 Comprehension results

Figure 10.1 shows the multiple-choice error rates of the five conditions in Experiment 8. Let's begin with the weak and strong transitive verbs. First of all, there is a clear main effect of noun plausibility. When the matrix subject is a good

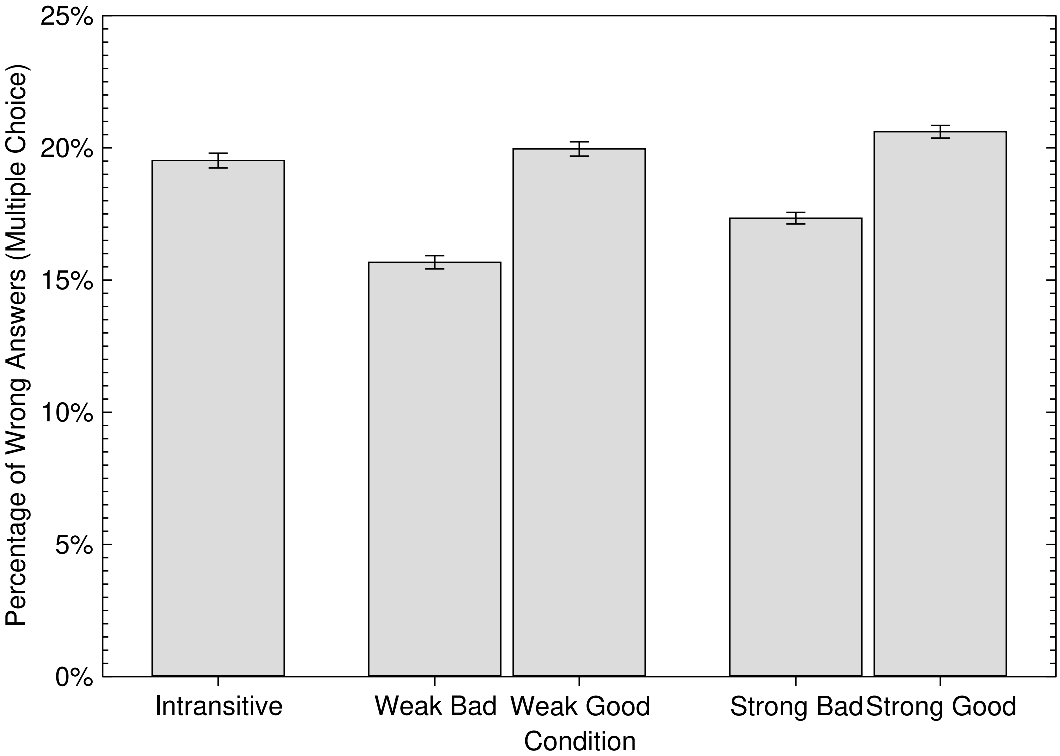


Figure 10.1: Multiple-choice comprehension error rate on the five conditions in Experiment 8.

object of the ambiguous verb, error rates are higher.[1] There is also a main effect of transitive frequency between the weak and strong conditions. The strong verbs, which encourage the incorrect transitive reading, lead to higher error rates.[2] In pairwise comparisons among the four transitive conditions, all differences are significant at the $p<0.001$ level except for the comparison between the two good object conditions, which does not quite reach significance.[3]

All of the findings just mentioned match expectations, except perhaps for the lack of a strong difference between the WG and SG conditions. However, the poor comprehension performance on the intransitive condition is surprising. The error rate on intransitives was significantly lower than that for the SG condition,[4] but not lower than the WG condition.[5] We might expect the intransitives to be the easiest condition overall because intransitive verbs provide the least support for the transitive reading. It does not seem that overall frequency can adequately explain the difficulty of the intransitive condition, since the weak transitive verbs were less frequent both in intransitive form and overall.

The most viable explanation for this difficulty may be that it reflects the general problem the model apparently has with intransitive verbs. Because the majority of verbs are transitive and the model is forced to do all of its processing in discrete timesteps, the model may have adopted a strategy of using the start of the direct object, or whatever follows the verb, to finalize its processing of the verb. As we saw in Chapter 7, this leads to problems when a sentence ends in an intransitive verb, but not when that verb is followed by a modifier. We might conclude that this also leads to problems when any clause ends in an intransitive verb, even if the sentence continues.

It is informative to compare this comprehension performance with the question-answering performance following message encoding. Sentences may be difficult to understand because they are syntactically confusing or because they are semantically confusing. By examining the error rate following message encoding, we can isolate the semantic effects, and thus get a better understanding of the relative difficulty specifically introduced by syntax.

Figure 10.2 shows the multiple-choice error rate on the message encoding and decoding task. Note that the scale

[1] $F(1,100714)=231$, $p<0.001$
[2] $F(1,100714)=22.1$, $p<0.001$
[3] $F(1,50362)=3.21$, $p=0.073$
[4] $F(1,48574)=8.81$, $p=0.003$
[5] $F(1,41386)=1.32$, $p=0.251$

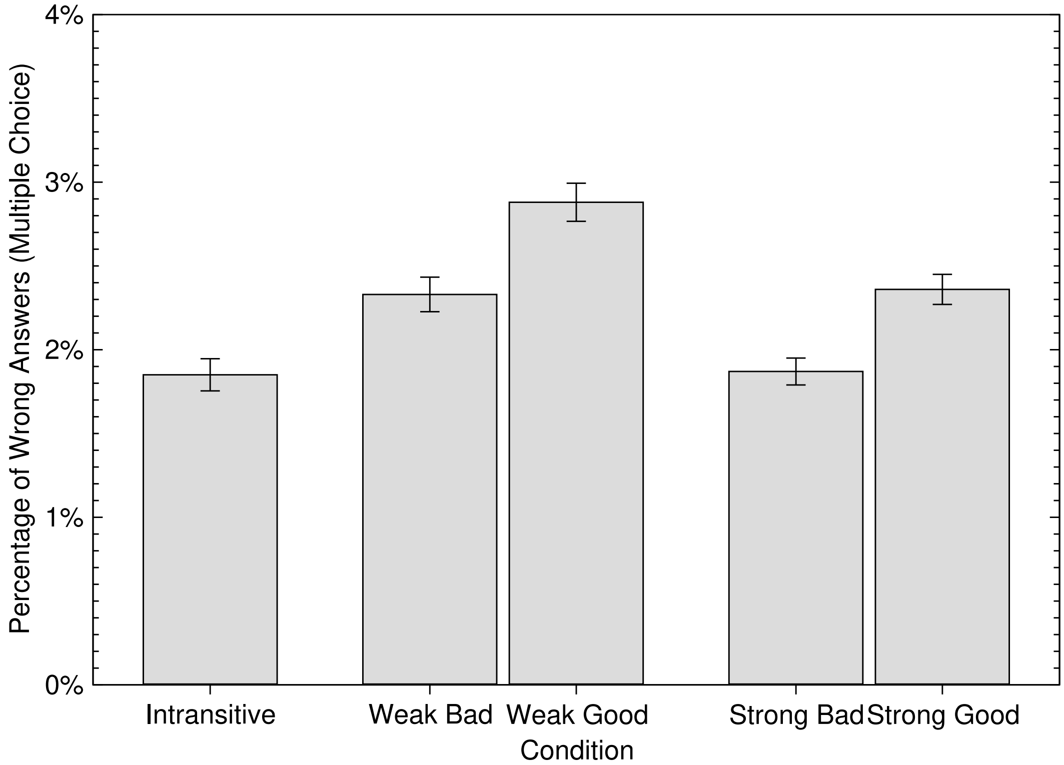


Figure 10.2: Multiple-choice encoding and decoding error rate on the five conditions in Experiment 8.

of the y-axis has changed and the error rates here are much lower than those resulting from comprehension. This indicates that much of the difficulty of these sentences is due to syntactic effects and not simply due to confusing semantic relations. Again, the good objects are harder than the bad objects. Thus, the difficulty of the good objects is at least partially explained by the more confusing semantic relationships associated with them. It is not clear how much of their effect during comprehension is due to an actual influence on syntactic processing mechanisms.

On the other hand, the pattern of difficulty across the verb-bias conditions is not the same for encoding as it was for comprehension. The strong transitives and the intransitives are easier than the weak transitives, which is opposite the pattern seen in comprehension. Therefore, the difficulty of the strong transitive and intransitive conditions in actual comprehension is due to syntactic, rather than purely semantic, effects. Probably, the differences between the weak and strong transitives in message encoding is due to the lower frequency of the weak transitives. If this confound were better controlled for, the advantage of the weak conditions in comprehension should be even more pronounced.

10.2.2 Reading time results

Figure 10.3 shows the SRT reading times for the model for the five conditions in Experiment 8. The intransitive condition is slower than the others on the first noun, before the verb is even reached. This suggests that the nouns selected as subjects of the intransitives were less plausible as subjects than those for the optionally transitive verbs. The slower reading in the intransitive condition continues until the article of the ambiguous NP. On this article there is a clear reverse effect of verb bias. The intransitive condition is slowest, followed by the weak and then the strong transitive conditions. On the noun, there is only a reverse effect of object plausibility. As should be expected, the good objects are faster than the bad ones. Interestingly, the intransitives, which have bad objects, are as fast as the transitives with good objects. If the model were expecting a direct object following the intransitives, one would expect that to be the slowest condition at this point.

The verb *was* (or *were*) is the disambiguation point in all conditions. Here we see two main effects. One is a verb bias effect, with the strongly transitive verbs slowest, followed by the weakly transitive and then the intransitive

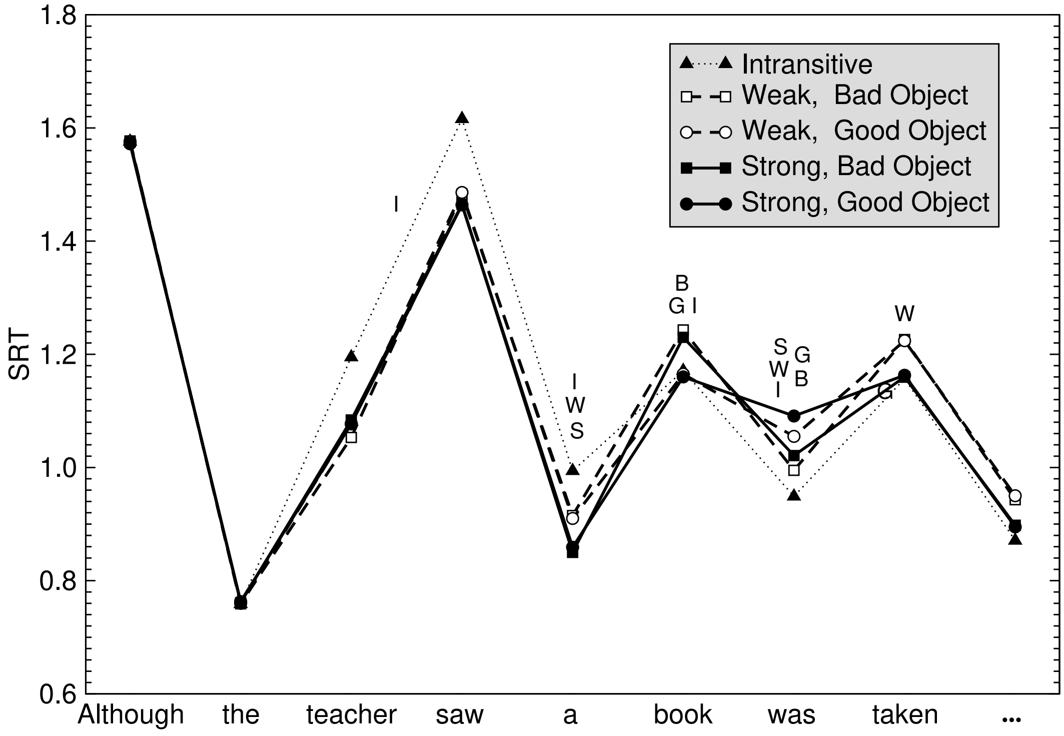


Figure 10.3: Reading times for the five conditions in Experiment 8.

verbs. The other one is an object plausibility effect, with the good objects being slower than the bad objects. Finally, at the main verb, *taken*, the weak transitives appear to be slower than the other conditions, with no effect of object plausibility. The reason for this is not clear. This may be some form of frequency effect, or it may be that the subjects of the main clause following weak transitives are, for some reason, more difficult than those following the other verbs.

The reading time results of the model on the ambiguous NP (*a book*) and at the disambiguation point (*was*) are shown in Table 10.1 along with the results of the empirical studies. The model appears to agree with all of the studies in terms of the relative reading rates of the conditions.

10.2.3 Summary and discussion

With the exception of the high error rate on the intransitive verbs, the model's comprehension performance on the NP/0 ambiguity was basically in line with expectations. Matrix subjects that are good objects of the preceding verb result in significantly higher error rates than bad objects. While, given the reading-time results, this is likely to be partially due to a syntactic effect, it is also partially attributable to the fact that the good object sentences are semantically more difficult, in the absence of any possible syntactic effects. Strong transitives, although more frequent and thus easier on a purely semantic basis, still led to higher error rates following comprehension. If frequency effects were better controlled for, the advantage for weak transitives in comprehension should become much stronger.

In terms of reading times, the model's performance replicates all of the experimental findings reviewed here, with the exception of the NP length effects of Frazier and Rayner (1982), which were not addressed. In the ambiguous region, intransitives are slower than transitives (Mitchell, 1987) and weak transitives are slower than strong transitives. Furthermore, bad objects are slower than good objects (Pickering & Traxler, 1998; Pickering et al., 2000). In the disambiguating region, all of these effects reverse. Strong transitives are slower than weak transitives (Mitchell & Holmes, 1985) and intransitives (Mitchell, 1987), and good objects are slower than bad objects (Pickering & Traxler, 1998; Pickering et al., 2000).

Despite these successes, it is worth mentioning that, in addition to the possible frequency confounds, there were some other factors that may have affected the results of this experiment. The weak transitive verbs often take only a

E	Although the mother asked the reporters were hit.
L	Although the mother asked the reporters the managers were hit.
E-AJ	Although the mother asked the mean reporters were hit.
L-AJ	Although the mother asked the mean reporters the managers were hit.
E-PP	Although the mother asked the reporters of the story were hit.
L-PP	Although the mother asked the reporters of the story the managers were hit.
E-MR	Although the mother asked the reporters that a father saw were hit.
L-MR	Although the mother asked the reporters that a father saw the managers were hit.
E-RR	Although the mother asked the reporters a father saw were hit.
L-RR	Although the mother asked the reporters a father saw the managers were hit.

Table 10.3: Sentence types used in the Ferreira & Henderson (1991) study and in Experiment 9. The syntactic structures of the sentences do not exactly conform to those used by Ferreira & Henderson.

limited set of objects. *Ate*, for example, is generally followed by a type of food. *Saw*, in contrast, could take a wide range of objects. The result is that the good objects for the weak transitives tend to be very good, in the sense that the verb is strongly associated with the noun, while the good objects for the strong transitives may be less preferred. Thus, the difference in reading time between the weak and strong transitives with good objects may be smaller than it should be. Eliminating this bias would have the effect of strengthening the weak/strong difference, which is already in the correct direction.

Some of the verbs, especially among the weak transitives and intransitives, frequently take sentential complements (*guessed*, *forgot*, *believed*, *thought*, *knew*, *hoped*, *asked*). Thus, when the ambiguous NP is encountered, there is actually a three-way ambiguity. The NP could be an object, the subject of the matrix clause, (59a), or the subject of an SC, (59b). The SC reading is only ruled out when the end of the sentence is reached. It is not clear what effect this confound may have had on the results of this experiment. Due to the limited number of verbs in Penglish, it is, unfortunately, unavoidable. However, items in other experiments should be checked to be sure SC-biased verbs are not used.

(59a) Before I realized[,] the deadline was upon me.

(59b) Before I realized the deadline was upon me, the thesis writing seemed to be progressing nicely.

10.3 Experiment 9

Ferreira and Henderson (1991) conducted a series of interesting experiments involving several variations on the basic early- and late-closure subordinate clause sentences. Table 10.3 lists a variety of sentence types which were used in one or more of the Ferreira and Henderson (1991) studies. Each of the sentence types in Table 10.3 is paired with a label, such as E-MR, which will be used to refer to it.

The task in all of the five experiments in Ferreira and Henderson (1991) was to decide whether or not the sentences were grammatical. In all but the second experiment, sentences were displayed using a word-at-a-time RSVP presentation, at a rate of 250 msec per word. In the second experiment, sentences were presented in a cumulative, segment-by-segment self-paced display. Several sentence types were tested in more than one of these experiments. In fact, all five experiments used versions of the E, L, E-MR, and L-MR types. Because the grammaticality ratings were quite consistent across experiments, it seems reasonable to simply average them and present a single set of results.

Figure 10.4 shows the average ungrammaticality ratings for the 10 sentence types used in the Ferreira and Henderson (1991) experiments, from least to most difficult. In order to maintain the convention that up is more difficult, the values displayed are actually 100 minus the grammaticality measures published by Ferreira and Henderson (1991). The simple late-closure sentences, L, were rated the most grammatical. The other variants on L were also rated quite grammatical, with the exception of the version containing a reduced object-relative clause, L-RR. It is not clear why the reduced relative should be deemed so much less grammatical than the marked relative in the late-closure cases, but of equal or slightly lower ungrammaticality in the early-closure cases. This is an unusual interaction. The sentences

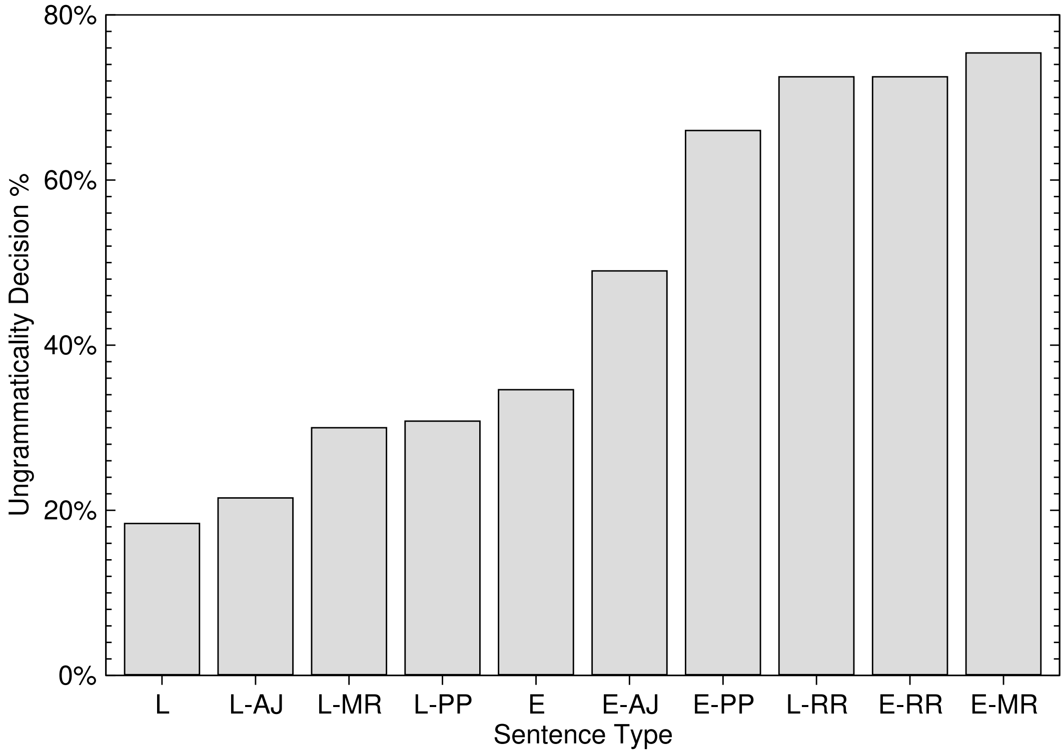


Figure 10.4: Experimental ungrammaticality decisions averaged across the five experiments in Ferreira & Henderson (1991). Examples of the sentence types are given in Table 10.3.

involving adjective modification of the ambiguous object, L-AJ, were nearly as good as L. The early-closure sentences, those starting with E, were deemed less grammatical than their late-closure counterparts. E-AJ was second to E, but the difference was greater than for the late-closure versions. E-AJ was rated significantly better than E-PP and the relative clause versions.

In order to evaluate the behavior of the CSCP model on such sentences, 50 examples of each type were generated. As in the examples shown in Table 10.3, corresponding sentences of different types were identical outside of the critical regions. Using a consistent set of items like this should reduce the variance and make for more reliable comparisons. There were, however, some potentially important differences in the structure of the sentences used by Ferreira and Henderson (1991) and that in the current study. Notably, Ferreira and Henderson used a pair of compound adjectives in the -AJ sentences to match the number of words in the -PP sentences. Because Penglish does not include compound adjectives, simple adjectives were used here.

One way to measure the complexity of these sentences, as we have done in the previous analyses, is to look at the simulated reading time at the disambiguation point, which in all cases was either *was* or *were*. Figure 10.5 shows these reading times in comparison to the ungrammaticality results obtained by Ferreira and Henderson (1991). The results seem to agree quite well. There is a clear effect of closure type, with the late-closure sentences easier than the early-closure ones. The relative clause sentences are much harder, particularly in the early-closure condition. However, there is a definite mismatch on two of the conditions, E-PP and L-RR. The model's reading times at disambiguation are not much affected by a prepositional phrase. Also, the model does not show the same interaction between closure type and the presence of a reduced relative as that seen in the Ferreira and Henderson results. In the model, the effect of a reduced relative is greater in the early-closure condition, rather than the late-closure condition. This seems to make more sense intuitively, and it will be interesting to see which type of interaction appears in future experiments. Overall, the correlation between the two patterns of results shown in Figure 10.5 is 0.72.

However, a better measure to compare with ungrammaticality decisions is not reading times at a particular point in the sentence, but the model's own ungrammaticality decisions. Section 6.5.5 described how ungrammaticality ratings and decisions can be derived when the model attempts to comprehend a sentence. In short, the ungrammaticality rating

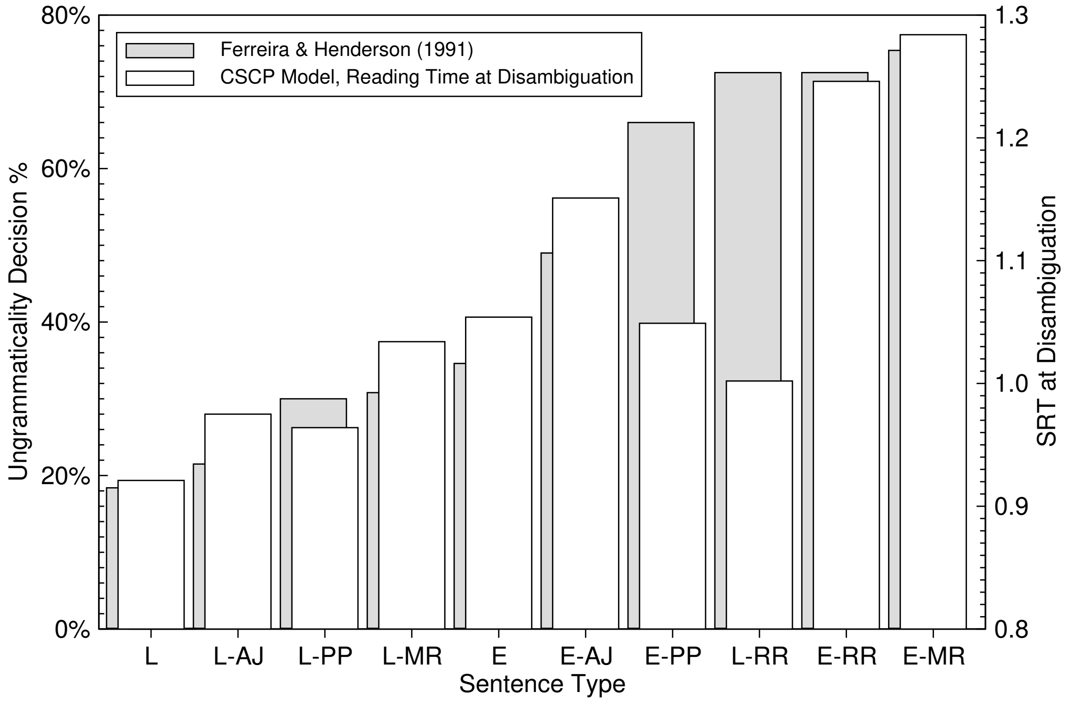


Figure 10.5: Simulated reading times at disambiguation of the CSCP model in comparison to experimental ungrammaticality decisions averaged across the five experiments in Ferreira & Henderson (1991). Examples of the sentence types are given in Table 10.3.

results from a function involving the product of two components, predictability and comprehensibility. The model's typical ungrammaticality ratings for actually grammatical sentences will range from close to zero for simple sentences to close to 10 for very complex ones. Because Ferreira and Henderson (1991) collected grammaticality decisions, rather than ratings, the simulated ungrammaticality decision (SUD) measure is the appropriate one for comparison. As explained in Section 6.5.5, the SUD was computed using a base criterion of 4 with noise in the range ± 2. Any sentence whose ungrammaticality rating falls above the noisy criterion is judged to be ungrammatical. For each sentence and each model, 100 decisions were made using different criteria and the results averaged.

Figure 10.6 shows the SUD for the CSCP model in comparison to the results of the Ferreira and Henderson (1991) study. Note that the Ferreira and Henderson data uses the scale on the left while the data from the model uses the scale at the right, which was adjusted to achieve better alignment. Clearly there are some major differences between the two sets of results. Primary among them is that the CSCP model considers the early-closure sentences to be as grammatical or more so than the corresponding late-closure sentences. Both the model and the subjects find the -AJ and -PP sentences to be slightly harder than their simpler counterparts, but the model finds the -PP sentences more grammatical, while the human subjects reported the opposite. Both the subjects and the model agree that the relative clause sentences are the hardest and that the effect of the addition of a relative clause is greater for the early-closure sentences. Notice, for example, that the model finds E, E-AJ, and E-PP to be quite grammatical, but the E-RR and E-MR to be rather ungrammatical. As in the reading-time data, the model does not find the L-RR sentences significantly harder than the L-MR, although Ferreira and Henderson did obtain that result. Overall the correlation between the model's results and the human data is still reasonably high, at 0.64.

10.3.1 Discussion

The fact that the model does not find the early-closure sentences to be less grammatical than the late-closure sentences is surprising but, in retrospect, does make some sense. The grammaticality rating is based on both the degree to which word predictions failed and on comprehension performance. It seems that both of these components favor the

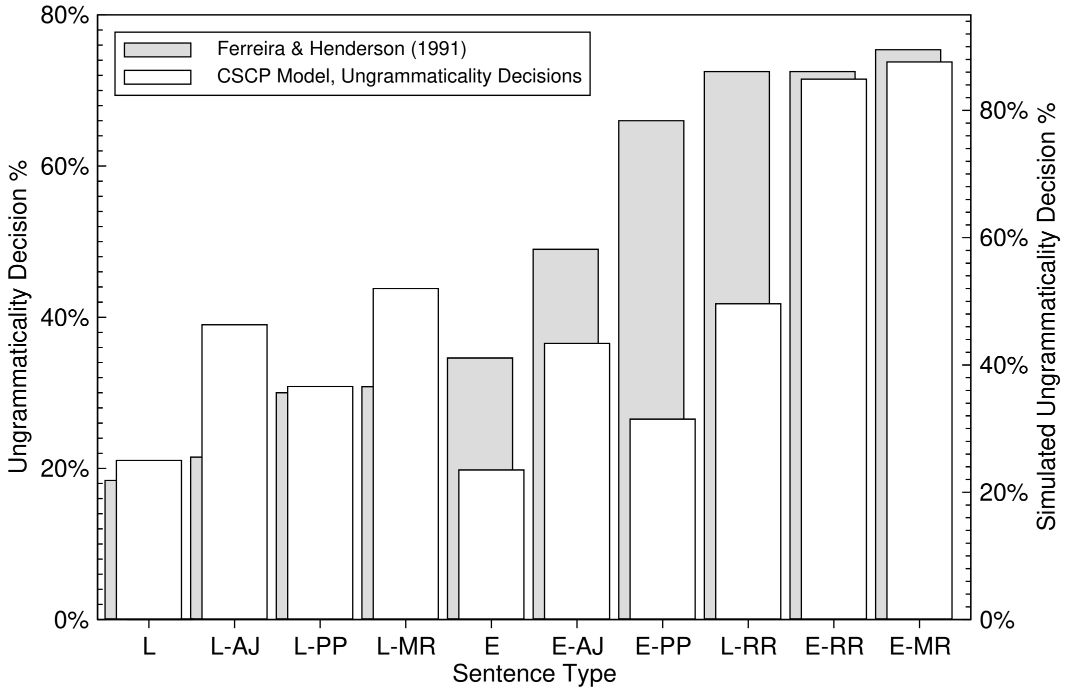


Figure 10.6: Simulated ungrammaticality decisions of the CSCP model in comparison to experimental ungrammaticality decisions averaged across the five experiments in Ferreira & Henderson (1991).

early-closure sentences. The semantic representations of early-closure sentences all have one fewer proposition than those of the late-closure sentences. Because the semantic system of the current model is rather limited in its ability to represent a large number of propositions, the addition of an extra proposition in the late-closure cases reduces their comprehensibility, which results in an impression of ungrammaticality. The manner in which the current sentences were generated may also have rendered the late-closure sentences more difficult. The late-closure sentences all involve the insertion of an extra NP (*the managers*) prior to the disambiguating verb and immediately following the ambiguous NP (*the reporters*). In the early-closure sentences, *the reporters* is the subject of the main clause and in the late-closure sentences, *the managers* is the subject. In order to balance the difficulty of the semantic relations in the main clause, these two nouns phrases were chosen to be very similar in meaning and identical in number. For example, they may both have been adult humans, or animals, or implements. The unfortunate side effect of this choice is that the two nouns are very confusable, which adds difficulty to the late-closure sentences. "The reporter was asked and the manager was hit or the reporter was hit and the manager was asked?" This added confusion, in the present grammaticality measure, results in a higher ungrammaticality rating for the late-closure sentences.

The second component of the ungrammaticality rating involves the worst pair of consecutive predictions, that is, over the course of the sentence, the pair of consecutive predictions that resulted in the most prediction error. If the worst prediction site occurred at the disambiguation point, one would expect the early-closure cases to be worse. And indeed it is true that the hardest predictions, more often for the early-closure sentences than they are for the late-closure sentences, are the two disambiguating verbs. However, in the majority of cases the worst predictions do not occur at disambiguation. The two predictions that are most often the hardest in the `E` and `L` sentences are those of the first noun and verb, *mother asked*. The reason for this may be that, at the start of the sentence, the model has no idea what may be coming, and predicting the first noun and verb are difficult. Predicting direct objects later in the sentence is easier because they are more highly constrained by the preceding context. The worst predictions for the late-closure sentences also sometimes involved the matrix subject, *managers*. This, too, is a difficult prediction because, in Penglish, the subject of a main clause is independent of the content of a preceding subordinate clause, so the model again has nothing to go on.

It seems that the model is not sufficiently garden-pathed on the early-closure sentences to overcome those aspects of the late-closure sentences that also contribute to a sense of ungrammaticality: more complex and confusable semantics and the difficulty of predicting an additional unconstrained NP. If the model were truly garden-pathed on the early-closure sentences, there would be a great deal of surprise at the disambiguation point, resulting in a large ungrammaticality rating.

Aside from the possibility that the CSCP model operates in a fundamentally different manner than the human sentence processing mechanism, there may be other explanations for the difference in grammaticality decision performance. One is that the method used to obtain the simulated grammaticality rating measure in the model is just not sensitive to the same factors as the mechanism used by human readers to assess grammaticality. Perhaps basing the grammaticality rating on word prediction and comprehension performance is not the right approach and we should seek a more direct measure that the model has detected a syntactic anomaly.

Alternately, it may be that the model is not strongly garden-pathed on the early-closure sentences because Penglish is not representative of English. In English, clause breaks are nearly always indicated by commas, when written, or, when spoken, by intonational phrase boundaries marked by pauses, pitch movement, and final word lengthening (Warren, Nolan, Grabe, & Holst, 1995). The reader or listener comes to rely on those cues as an aid to parsing and, when the cues are eliminated, garden-pathing is more likely. On the other hand, in Penglish there are no cues marking clause boundaries. Therefore, the model learns not to rely on them. When faced with a potential ambiguity, it is less likely to commit to the NP reading due to the lack of a standard boundary.

In order to test this hypothesis, the Penglish language should be changed so that clause boundaries, in the form of a special word indicating a comma or pause, are provided in the majority of cases. The prediction is that, in the absence of such a marker, the model would show a much stronger garden-path effect than it does presently. This should be evidenced in both longer reading times at disambiguation and higher ungrammaticality ratings for early-closure sentences.

10.4 Experiment 10: Incomplete reanalysis

Christianson, Hollingworth, Halliwell, and Ferreira (2001) (see also Ferreira, Christianson, & Hollingworth, 2001) questioned the extent to which reanalysis on garden-path sentences, like those we have examined here, is fully completed. When subjects read a sentence such as (60a), their initial impression is likely to be that the man is hunting the deer. By the time they have reached the end of the sentence, they should have recovered to the extent that they now believe that the deer ran into the woods. But is there a remnant of the initial misanalysis? Will they still agree to the statement that the man was hunting the deer? In short, that does appear to be the case.

(60a) While the man hunted the deer ran into the woods.

(60b) While the man hunted the deer paced in the zoo.

(60c) While the man hunted the pheasant the deer ran into the woods.

In their first experiment, Christianson et al. (2001) asked subjects to read sentences like (60a)–(60c) and to answer the true/false question corresponding to, "Did the man hunt the deer?" After reading (60a), subjects are expected to answer "yes" quite frequently, if they have failed to completely reanalyze. Condition (60c) is a late-closure version for which the subjects are expected to answer "no." In sentence (60b), the fact that the deer is in a zoo is supposed to make it implausible that the man is hunting the deer, so subjects may again be less likely to answer "yes." In addition to the conditions shown above, which are the short versions, each sentence also had a long version in which the ambiguous noun, *deer*, was post-modified by a relative clause with a compound predicate adjective, such as "*that was brown and graceful*."

Figure 10.7 shows, in the gray bars and using the scale on the left, the frequency with which subjects responded "yes" to the question. Christianson et al. actually conducted two versions of the experiment, one with RSVP presentation and the other allowing subjects to self-pace themselves and view the whole sentence. Similar effects were obtained, so I have averaged the results to produce those shown in the figure. As expected, the early-closure sentences, like (60a), resulted in more "yes" answers than did the late-closure sentences. Those with relative clauses following the noun also resulted in higher error rates. Although I have not shown the results for the implausible condition, (60b), suffice it to say that the error rate was similar to the late-closure sentences when short and fell between (60a) and (60c) when long. Thus, it appears that readers frequently do fail to completely revise their initial interpretations when those interpretations are not contradicted by additional information about the situation.

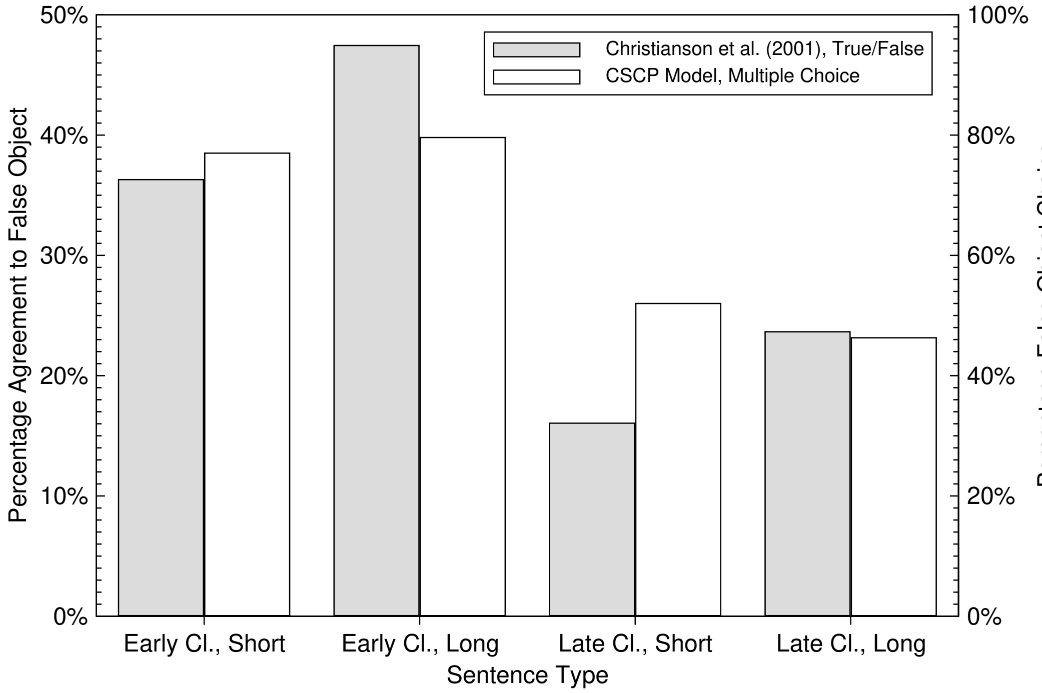


Figure 10.7: Frequency with which subjects in Christianson et al. (2001), Expt. 1, or the CSCP model agreed to the false proposition that, "The man hunted the deer," in sentences such as (60a) and (60c). Human data is based on true/false questions and uses the scale at the left and data from the model is based on multiple-choice responses and uses the scale on the right.

10.4.1 Experiment 10a

Even before learning of the Christianson et al. (2001) results, I suspected that the CSCP model would be subject to a similar form of confusion. So, of course, I attempted to evaluate the model on similar sets of items. Early and late-closure sentences were constructed that were fairly similar in structure to those used by Christianson et al.. Like most of their items, the main clause contained an active, intransitive verb with a verb-modifying PP. Unfortunately, Penglish does not allow compound adjectives or predicate adjective relative clauses. Ordinary subject-relatives could have been used instead, but they are significantly more complex, syntactically and semantically. Therefore, in the long conditions the ambiguous noun was post-modified by a PP, rather than a relative clause. The PP is hopefully more similar in complexity to the phrases used by Christianson et al. (2001). Each condition contained 180 sentences.

The implausible conditions, such as (60b) could not be recreated in this experiment. The reason is that, in Penglish, there is no semantic connection between a subordinate and a matrix clause. The two are generated independently. That is, whether the deer is running in the woods or buying a latte at Starbucks has no bearing on the plausibility of its being hunted. Thus, the sentences used to test the model probably fall somewhere between the plausible and implausible conditions used by Christianson et al.. The matrix clause neither encourages nor discourages the garden-path reading.

The model also cannot be asked true/false questions. Therefore, it was instead asked the fill-in-the-blank question, "Who was the object of the hunting?" The multiple-choice criterion was used to assess the response. If the response was closer to "the deer" than to any of four other distractors, then the model presumably would have responded "yes" to the question, "Did the man hunt the deer?" In the white bars, Figure 10.7 shows the frequency with which the model answered "the deer" in each of the four conditions, using the scale on the right. The most important effect was replicated: the frequency of confusion is higher for the early-closure sentences than for the late-closure sentences. It is interesting that for both the model and the subjects there were still a substantial percentage of incorrect answers in the late-closure conditions.

The additional confusion in the long sentence conditions was not adequately replicated. For the early-closure sentences, the added PP resulted in slightly greater confusion. But for the late-closure sentences, the added PP resulted in significantly less confusion. The reason may have to do with the fact that multiple-choice questions were used on the model. The distractors in such questions are all nouns, including those appearing elsewhere in the sentence. The added PP provides an extra noun that may be an attractive distractor, reducing the frequency with which "the deer" is chosen in either the early or late-closure sentences, thus countering any influences of a post-modifier that may have gone in the opposite direction. In contrast, the predicate adjective relative clauses used by Christianson et al. did not contain any nouns that may have influenced subjects' interpretations or true/false decisions. One additional difference between the results from the humans and the model is that the error rates for the model are about twice those of the humans. Aside from the possibility that the model is simply more prone to confusion on such a task, the difference may be partially due to the use of a multiple-choice rather than a true/false measure. Interestingly, their second experiment, to be discussed next, resulted in much higher error rates, more on par with those from the model.

10.4.2 Experiment 10b

The astute reader should have realized that there is an obvious problem with the first experiment conducted by Christianson et al. (2001). In a sentence such as, "While the man hunted, the deer ran into the woods," it may still be true that the man is hunting the deer, whether or not it is explicitly stated. Therefore, even if they were not garden-pathed, subjects might be justified in their assumption that the man was hunting the deer. Their second experiment represents one attempt to control for this possibility. In this case, sentences like (60d)–(60g) were used.

(60d) As Harry chewed the brown and juicy steak fell to the floor.

(60e) As Harry chewed the steak that was brown and juicy fell to the floor.

(60f) The brown and juicy steak fell to the floor as Harry chewed.

(60g) The steak that was brown and juicy fell to the floor as Harry chewed.

Sentence (60e) is similar to the long, early-closure condition used before and will be known as the post-modifier condition. Sentence (60d) uses adjective modification prior to the ambiguous noun, and will thus be known as the pre-modifier condition. In (60f) and (60g) the clause order has been reversed to eliminate the ambiguity. If subjects are concluding that Harry is chewing the steak purely due to its semantic plausibility and not because of a misparsing, the error rates should be identical between (60d) and (60f) and between (60e) and (60g). Any difference can be attributed to a syntactic effect. As another form of control, subjects were also asked a question about the matrix clause, "Did the steak fall to the floor?" If error rates on this question are as high as those for the subordinate clause question, it would indicate that subjects are generally confused, not necessarily that they are misparsing and only partially reanalyzing.

Figure 10.8 gives the results of this second experiment. `Pre` and `Post` indicate pre-modification (head late) versus post-modification (head early). `SM` is the ambiguous subordinate-first ordering and `MS` is matrix-first. The bars on the left half reflect the percentage of incorrect "yes" answers to the subordinate clause question. The bars on the right reflect the percentage of incorrect "no" answers on the matrix clause question. To begin with, error rates on the matrix clause question are much lower than those on the subordinate clause question, indicating that subjects are not just generally confused. Error rates are also lower for the matrix-first sentences, although that is really only true for the post-modifier condition. Higher error rates were obtained in the post-modifier condition than in the pre-modifier condition. This was particularly true for the `SM` sentences, but there were also small, though perhaps not significant, effects for the `MS` sentences.

The model was tested using similar sets of sentences. Again, PPs were used in place of the post-modifying RCs. Because Penglish does not have compound adjectives, single adjectives were used in the pre-modifier condition. In place of the true/false matrix sentence question, the model was asked to answer a multiple-choice question such as, "What fell?" The results are shown in the white bars in Figure 10.8. Like the subjects, the model is much better at answering the matrix clause questions than the subordinate clause questions. The error rate on the matrix clause questions is somewhat higher for the model than for the subjects. However, the model is answering multiple-choice questions while the subjects are answering easier true/false questions. In answering the matrix clause questions, the model also shows an advantage for the pre-modification sentences. This is probably due to the slightly greater syntactic complexity of the post-modifier conditions.

In answering the subordinate clause question, the model also shows a clear difference between the `SM` and `MS` conditions. Although it is still confused quite often on the `MS` sentences, it is confused far more often by the garden-

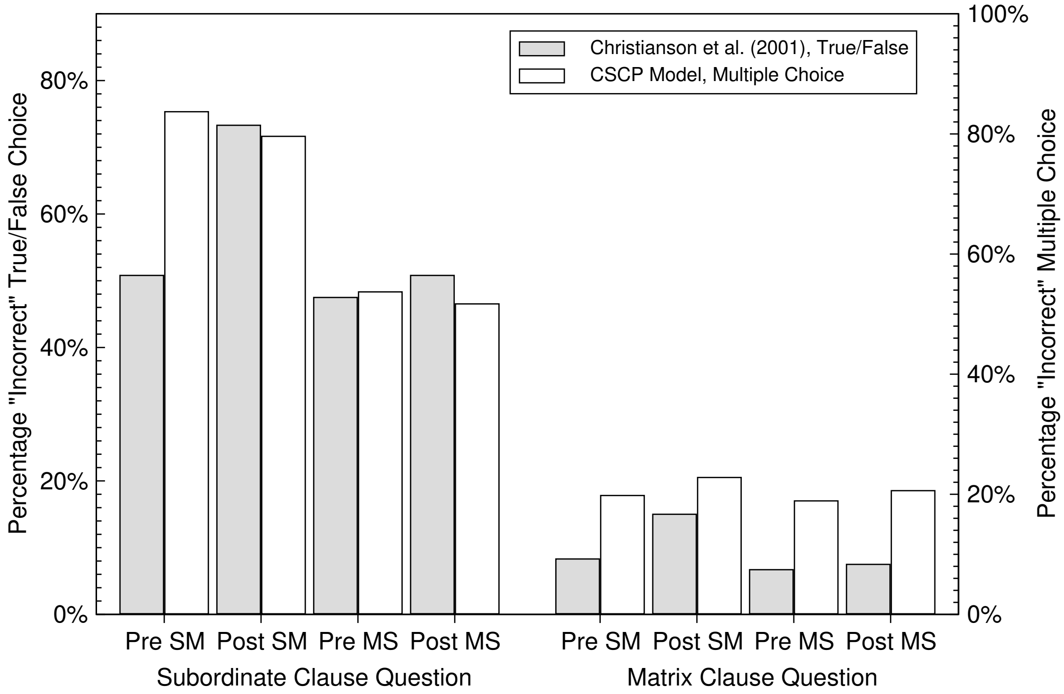


Figure 10.8: Frequency with which subjects in Christianson et al. (2001), Expt. 2, or the CSCP model agreed to the false subordinate clause proposition that, "The man hunted the deer," or the false matrix clause proposition, "The deer ran into the woods." Human data (gray bars) is based on true/false questions and uses the scale at the left and data from the model (white bars) is based on multiple-choice responses and uses the scale on the right. Pre means pre-modified or head late, Post means post-modified or head early. SM is subordinate clause first and MS is matrix clause first.

pathing SM sentences. Unlike the human subjects, however, the model actually has a slightly lower false positive rate on the post-modification sentences than on the pre-modification ones. Again, this may be due to the additional attractive distractor that appears in the post-modification condition. The strongest disagreement between Christianson et al.'s results and those of the model is on the Pre SM condition. This is the hardest condition for the model, but for the subjects it is not much harder than the MS conditions. It actually seems, intuitively, that the model's results make more sense. If subjects are indeed being mislead largely by the temporary ambiguity, then the Pre SM condition should result in significantly more false positives than the Pre MS condition, whereas the data show only an insignificant increase.

As the final and strongest control for the possibility that subjects believe the man was hunting the deer or eating the steak purely on the basis of its semantic plausibility, Christianson et al. (2001) conducted a third experiment in which they compared ordinary optionally transitive verbs, like *hunted*, with *reflexive absolute transitive* (RAT) verbs, such as *dressed*. The RAT verbs have the property that, in their intransitive form, they are obligatorily reflexive. Thus, if such a verb is interpreted intransitively, adopting the false transitive reading would create a contradiction. Indeed, they found that given the MS clause order, subjects were much more likely to respond "yes" to the subordinate clause question for the standard optionally transitive verbs than for the RAT verbs. But with the SM word order, subjects still responded "yes" over half the time for the RAT verbs. This shows that the temporary ambiguity has a lasting effect on readers' interpretations even when the false transitive reading is incongruous with the intransitive reading. Unfortunately, this result cannot be addressed with the model because Penglish does not include RAT verbs.

Experiment 10 has shown that, like human readers, the model is prone to incomplete semantic reanalysis when it recovers from a garden-path sentence. Although, for optionally transitive verbs, both humans and the model are prone to agree to a transitive reading of the subordinate clause even with the MS word order, they are more likely to do so under the SM word order. This indicates that the model, like us, is influenced both by syntactic ambiguity and by semantic influences that are unsupported by syntax. It is likely that the same would not be true of models that are heavily dependent on syntax.

Because Penglish does not include all of the necessary structures and because the model cannot be asked true/false questions, a truly faithful replication of these experiments was not possible. The use of multiple-choice responses, coupled with the fact that the PPs contained an additional NP while the post-modifiers used by Christianson et al. (2001) did not, may have contributed to the failure of the model to replicate the fact that garden-path readings are more often retained in the post-modifier conditions.

10.5 Summary and discussion

The CSCP model is able to replicate many of the behavioral effects evidenced by human readers in processing the subordinate clause, or NP/0, ambiguity. Experiment 8 showed that more strongly transitive verbs lead to faster reading times on the ambiguous NP but slower reading times at disambiguation. Good objects also lead to faster reading of the object and slower reading at disambiguation. Despite the appropriate reading-time results, the model did have some trouble comprehending sentences with intransitive verbs. I suspect that this may be due to a strategy the model has adopted to help overcome the processing limitations that have been placed on it. One unavoidable confound in this experiment was the use of verbs that frequently take sentential complements, resulting in a three-way ambiguity. Researchers preparing experimental items should be aware of this possibility.

Experiment 9 measured grammaticality decisions on a variety of early- and late-closure sentences studied by Ferreira and Henderson (1991). Using reading times, the model was able to replicate most of the findings for relative sentence difficulty, with the exception of the high ungrammaticality ratings obtained by Ferreira and Henderson in the E-PP and L-RR conditions. It would be surprising if ungrammaticality of the L-RR condition, which was far above the L-MR and other late-closure conditions, could be replicated. Using ungrammaticality decisions, the model replicated the fact that the E-RR and E-MR conditions were extremely hard, but failed to replicate the fact that the other early-closure sentences should be considered less grammatical than the corresponding late-closure varieties. I have suggested that the reason for this may be that the model does not experience a sufficiently strong garden-path effect, in the absence of a relative clause, because it was never exposed to helpful clause boundaries during training. If Penglish, like English, normally marked the clause boundary with a comma or intonational boundary, the model would probably experience a greater garden-path effect in the absence of such a marker.

Experiment 10 demonstrated that the model, like humans, often fails to fully reanalyze the initial interpretation in early-closure sentences. Both humans and the model are sensitive to semantic and syntactic influences in the extent to which the incorrect transitive interpretation is adopted.

Some additional experimental findings on the subordinate clause ambiguity were not addressed here but are worth discussing briefly. Stowe (1989) examined the effect of animacy of the subordinate clause subject on the NP/0 ambiguity. Inanimate subjects were expected to provide less support for the incorrect transitive reading. In addition to manipulating subject animacy, sentences were either ambiguous or disambiguated with an adverbial phrase following the first verb. Indeed, in a self-paced word-by-word grammaticality decision task, subjects showed an ambiguity effect in reading times to the disambiguating verb following animate subjects, but no such effect following inanimate subjects. These results suggest that human subjects can use plausibility information in selecting the initial analysis of an ambiguity such as this.

Clifton (1993) performed a similar experiment using eye-tracked reading and controlling for semantic plausibility across the conditions. The unambiguous conditions in this case were disambiguated using a comma, rather than an adverbial phrase. To his apparent surprise, Clifton replicated Stowe's findings. There was again an ambiguity effect on the disambiguating word for animate subjects, but not for inanimate ones. Clifton argued that these effects must be due to reanalysis, and not to first-pass preferences on the basis of an ambiguity effect on the ambiguous NP which appeared for both inanimate and animate subjects. This latter finding, Clifton suggested, shows that subjects were adopting an initial transitive reading after the inanimate subjects. However, slower reading on the NP for the ambiguous condition could just as well reflect preparation for the upcoming main clause. In the disambiguated condition, this preparation could have started earlier, when the comma appeared. It does not seem that the Clifton (1993) finding provides definitive support for one parsing theory over another, but the replication of Stowe's result is interesting. I expect that the CSCP model would show a similar sensitivity to the animacy of the subordinate-clause subject. It may be possible to conduct such an experiment, although the number of items that can be generated using inanimate subjects might be too small in the current version of Penglish for this to be practical.

Chapter 11

Relative Clauses

Having considered several types of temporarily ambiguous sentences, we turn to a generally unambiguous construction that nevertheless provides a potentially valuable test of sentence comprehension systems: the relative clause. Although much of the empirical work on RCs has involved heavily nested uses, this chapter will focus primarily on single relative clauses. Two main factors are at work in single RC sentences, whether the RC is center-embedded or right-branching and whether it is subject-relative (a.k.a. subject-extracted or subject-focus) or object-relative. Examples (61a)–(61d), repeated from (18a)–(18d), list the four basic sentence types. The symbols C and R will denote center-embedding versus right-branching and the symbols S and O will denote subject- and object-relatives.

(61a) CS: The dog that bit the cat chased the bird.

(61b) CO: The dog that the cat bit chased the bird.

(61c) RS: The dog bit the cat that chased the bird.

(61d) RO: The dog bit the cat that the bird chased.

The main verb/reduced relative, which was discussed in Chapter 8, involved reduced center-embedded object-relatives. Here we will generally be looking only at marked forms of the clauses. One other variable to consider is the degree to which sentences are semantically confusable. In (62a), the semantic relationships are clearly defined. Among the nouns, *cops* is the most likely agent of *arrested* and *hoodlum* is the most likely patient. Likewise, cops are unlikely to be riding a single bicycle, let alone a hoodlum. Well, perhaps only metaphorically. In contrast, each of the nouns in (62b) could fill any of the thematic roles, increasing the possibility that confusion will occur. In such a case, it is expected that subjects will have more difficulty remembering who did what to whom.

(62a) The cops arrested the hoodlum that rode a bicycle.

(62b) The lawyer met the accountant that mentioned the economist.

Other factors have been studied in connection with single RCs, including effects of discourse and the distinction between restrictive and non-restrictive relatives (Grodner, Gibson, & Watson, in press; Gibson et al., in press), but they will not be covered here.

11.1 Empirical results

Blaubergs and Braine (1974) and Marks (1968) tested question answering and grammaticality ratings of sentences with from one to five CO or RS embeddings. Possibly because they included so many difficult sentences, these experiments found no reliable differences between the single CO and RS sentences.

Holmes (1973) used RSVP presentation and measured sentence recall as an indicator of comprehensibility. Included among the items were center-embedded and right-branching RCs. Both sets were evenly mixed between subject- and object-relatives. Holmes found that the CS/CO sentences were significantly easier to recall than the RS/RO, with 8.46 versus 7.42 mean words recalled. However, the sentences were not equated for meaning and the center-embedded sentences were judged to be slightly more natural. It is not clear if this should be considered a confounding or an explanatory factor. Table 11.1 provides a summary of these and the other results discussed here.

Baird and Koslick (1974) tested the four RC sentence types using auditory presentation and fill-in-the-blank questions. All nouns were humans, and non-associated noun/verb pairs were used, so the semantic relations were highly confusable. Error rates for the subject-extraction sentences were significantly lower than those for the object-relatives. There was a hint of an interaction, with the CS condition the easiest and the CO condition the hardest, but it was probably not significant.

Hakes et al. (1976) asked subjects to listen to sentences and perform a phoneme monitoring task followed by paraphrasing. Their first experiment contrasted CO and RO sentence types, but the sentences were all complex in ways other than the relative clauses, including other main and subordinate clause types. The paraphrasing error rate was 54.9% for CO sentences and 59.1% for RO sentences. The second experiment was similar to the first except that subject-relatives were used and embedded verbs were in the past progressive tense (*[who were] running*) to permit reduction. Again the unreduced case revealed a significant advantage for center-embedding with a paraphrasing error rate of 32.1% for CS versus 37.5% for RS. Although a statistical test was not done, it seems reasonable to assume that the subject-relatives were easier than the object-relatives.

Hudgins and Cullinan (1978) studied single RC sentences using a sentence-elicited imitation task. Subjects listened to a recorded sentence and then tried to repeat it verbatim. Based on error rate, this study found a significant advantage for subject-relatives over object-relatives, both for short sentences and for longer sentences involving adverbials and other filler material. There was also a consistent advantage for center-embedded over right-branching clauses, but this was not a significant difference in all cases. Error rates for the short sentences were CS:14%, RS:20%, CO:38%, and RO:40%.

Ford (1983) used word-by-word lexical decision as a measure of reading difficulty. Only CO and CS sentences were compared and were arranged in pairs matched for the words used in the relative clauses. As a result, the relative clauses in the two conditions were roughly opposite in meaning, as in Hakes et al. (1976). Lexical decision reaction times were significantly longer for object-relative sentences only on the relative clause verb, the main clause verb, and the main object determiner. This once again confirms that single object-relatives are more difficult than single subject-relatives when modifying the matrix subject.

King and Just (1991) performed a self-paced reading study in which subjects had to remember the last word in each of a series of three sentences. The CO and CS sentences under investigation appeared as the second or third in the series. Reading times were significantly longer during the relative clause in the object-relative sentences.

Finally, Gibson, Desmet, Watson, Grodner, and Ko (in press) conducted two self-paced reading experiments on singly-embedded RCs. In the first, CS, RS, CO, and RO sentences were tested using self-paced reading and true/false question answering. In one set of conditions, the sentences were in the usual form, while in another set of conditions the sentences were embedded within a sentential complement. We will ignore the embedded condition for now. The words in the RC were held constant but the word order, and thus the meaning, differed between the subject- and object-relatives. A rating study showed that the semantic relation in the subject-relatives was non-significantly more plausible than that in the object-relatives (3.58 vs. 3.49). Gibson et al. found two main effects: center-embedded RCs were easier to read and comprehend than right-branching ones and subject-relatives were easier to read and comprehend than object-relatives, although these effects were only significant in reading times.

The results of these studies seem quite clear. As summarized in Table 11.1, they consistently find that center-embedding is easier than right branching and that subject-relatives are easier than object-relatives. As these main effects would imply, the CS condition is consistently the easiest. The RO condition is the hardest in all experiments except Baird and Koslick (1974), in which it was a close second to CO. Whether the RS or CO condition is easier varies from one study to the next.

11.2 Experiment 11

A predecessor to the CSCP model was trained on a much simpler language than Penglish and tested on the four RC sentence types. Although reading times were not measured, its comprehension performance was very much in line with the empirical results just reviewed. There was a large effect of RC-type and a small effect of location. Average

Study	Input Format	Test Method	Results
H73	RSVP	recall	C $<$ R
BK74	auditory	FITB questions	CS $\leq$ RS $<$ RO $\leq$ CO
HEB76	auditory	recall	CS $<$ RS $<$ CO $<$ RO
HC78	auditory	recall	CS $<$ RS $<$ CO $\leq$ RO
F83	self-paced	lexical decision	CS $<$ CO
KJ91	eye-tracking (ML)	reading time	CS $<$ CO
GDWGK02	self-paced	T/F questions	CS $\leq$ CO $\leq$ RS $\leq$ RO
GDWGK02	self-paced	reading time	CS $<$ CO $\leq$ RS $<$ RO
Early Model	RSVP/self-paced	FITB questions	CS $<$ RS $<$ CO $<$ RO
CSCP Model	RSVP/self-paced	FITB questions	RO $\leq$ RS $<$ CS $<$ CO
CSCP Model	RSVP/self-paced	reading time	RO $<$ CO $=$ RS $\leq$ CS

Table 11.1: Summary of qualitative results on single RC sentences. C = center-embedded, R = right-branching, S = subject-extracted, O = object-extracted, $<$ indicates the condition on the left is probably significantly faster, $\leq$ indicates a smaller difference, probably non-significant, $=$ indicates no appreciable difference, FITB = fill-in-the-blank, T/F = true/false, ML = memory load.

strict error rates were 8.0% for CS, 13.3% for RS, 20.3% for CO and 27.7% for RO. The subject-extraction advantage is attributable to frequency, as RCs containing them were about 7 times more common than object-extraction RCs. However, the results are not entirely explained by frequency because the RO sentences were actually slightly more common than the CO. The fact that the model produced such nice results, especially since, at the time, I was not aware that they matched the empirical findings, was rather exciting. However, the statistics in the language on which that model was trained were not carefully controlled and the language did not include many of the other complexities found in Penglish. Importantly, the manner in which semantic propositions were encoded in that language also differed from the current method, but more on that later.

11.2.1 Experiment 11a

Along with training on a language with appropriate syntax statistics, an important control in performing a valid comparison of the difficulty of relative-clause types is that the semantic relationships in the sentences are balanced for plausibility. This was not sufficiently done in any of the empirical experiments reviewed here. For example, because they were primarily interested in reading times, Gibson et al. (in press) used identical words within the subject- and object-relative clauses with the result that the clauses were essentially opposite in meaning. Likewise, the matrix clauses had opposite meanings between the center-embedded and right-branching cases. While plausibility ratings were acquired to try to rule out a large advantage for one condition over another, such ratings may not be sensitive enough to detect potentially real differences in semantic plausibility.

Therefore, the first test of single-relative-clause sentences used materials with fully counterbalanced semantic relationships, which were constructed as follows: 20 word-sets were generated, each containing two transitive verbs and three nouns, which were either singular or plural. Each set of words can be rearranged to form 12 different sentences of each of the four types. Table 11.2 shows the 12 CS sentences that can be constructed from the verbs *asked* and *believed* and the nouns *lawyer*, *cop*, *fathers*. These words could also be arranged to form 12 CO, RS, or RO sentences. Among these twelve sentences, for any of the sentence types, there are 24 “actions,” with two occurrences of each type of action. For example, there are two cases of “lawyer asked cop,” two cases of “lawyer asked fathers,” two cases of “fathers believed lawyer,” etc. Therefore, summing across all twelve sentences, the plausibility of the SVO relationships should be balanced across the sentence structures. The 20 word-sets times 12 sentences per set gives 240 total sentences per type in this experiment.

The multiple-choice question-answering error rates of the model on the four RC sentence types are shown in Figure 11.1. The white bars represent the comprehension performance. That is, the model was fed the actual sentence and then asked questions about the message it had derived. Sadly, the results do not match the empirical findings. The CO condition is quite difficult, as it should be. However, the RO condition, which should be the hardest or nearly

The lawyer that asked the cop believed the fathers.
The lawyer that asked the fathers believed the cop.
The cop that asked the lawyer believed the fathers.
The cop that asked the fathers believed the lawyer.
The fathers that asked the lawyer believed the cop.
The fathers that asked the cop believed the lawyer.
The lawyer that believed the cop asked the fathers.
The lawyer that believed the fathers asked the cop.
The cop that believed the lawyer asked the fathers.
The cop that believed the fathers asked the lawyer.
The fathers that believed the lawyer asked the cop.
The fathers that believed the cop asked the lawyer.

Table 11.2: The complete set of 12 CS sentences composable from the verbs *asked* and *believed* and the nouns *lawyer*, *cop*, and *fathers*.

so, is actually the easiest and the CS condition, which should be the easiest, is the second hardest. Whereas human subjects and the previous version of the model have a preference for center-embedding, the current model has a strong preference for right branching. There also seems to be an interaction between RC type and location, such that CO is the hardest, but RO is the easiest, although the latter is not actually statistically different from RS.[1]

So the question, then, is why does the model not perform like the human subjects, or even like its predecessor? Two main factors contribute to the difficulty of comprehending a sentence: its syntactic complexity and its semantic complexity. Semantic complexity is too often overlooked in discussions of sentence processing, but it may be critical to overall sentence difficulty and must be treated seriously. Although we have attempted to control for the semantic plausibility of the clauses, taken independently, across the sentence types, there may be interactions between the propositions in the two clauses that result in more or less semantic complexity for an individual sentence as a whole.

But before considering these factors directly, we should first talk about our old friend, frequency. While not a direct factor in itself, frequency differences can affect the syntactic and semantic complexity of sentence structures, as the model will tend to adapt itself to the more frequent types of input. In both Penglish (Section 5.2.5) and English (Section 4.3), subject-relatives are more common than object relatives and object-modifying RCs are more common than subject-modifying (center-embedded). Thus, the RS condition is the most frequent, with sentences of this form appearing about 1,273 times per million in Penglish. RO and CS are essentially tied with frequencies of 621 and 601, respectively, and CO is by far the least frequent, at 64 occurrences per million. Thus, frequency is probably one contributor to the fact that the CO condition is the hardest, but it does not explain the advantage of RO over RS or CS.

Turning back to the question of syntactic versus semantic effects, we can begin to tease apart the separate contributions of these two factors by examining the performance of the semantic system by itself. If the propositions that compose the meaning of a sentence are fed into the encoder network (see Section 6.2) and the resulting message is queried directly, we can measure the difficulty of semantic decoding in isolation from syntax. The white bars in Figure 11.1 show the multiple-choice error rates of just the semantic system. Note that the error rates for the semantic system parallel those of the full comprehension system, in the gray bars. The same relative ordering of conditions is maintained, although the difference between CS and CO is smaller and the difference between RS and RO is larger. This suggests that the comprehension error rates are largely attributable to semantic encoding.

The black bars in Figure 11.1 show the difference between the gray and white bars. Known as the *comprehension-induced error* (CIE), this is the residual error if the semantic error is subtracted from the overall comprehension error. This error may be attributable to syntactic processing.[2] Note that the results are beginning to look a bit more reasonable. Although there is still a preference for right-branching sentences, there is now a consistent preference for

[1] F(1,17278)=1.881, p=0.170

[2] The subtraction of semantic error from comprehension error may not directly reflect syntactic difficulty, because the two rely on the same decoder system and are not really independent. As discussed in Section 7.4.1, a small syntactic error and a small semantic error may combine to produce a large comprehension error. Or if the decoder is robust, it may recover from a large syntactic processing error to produce only a small decrease in comprehension performance. Thus, such a subtraction is, at best, a noisy indicator of syntactic difficulty.

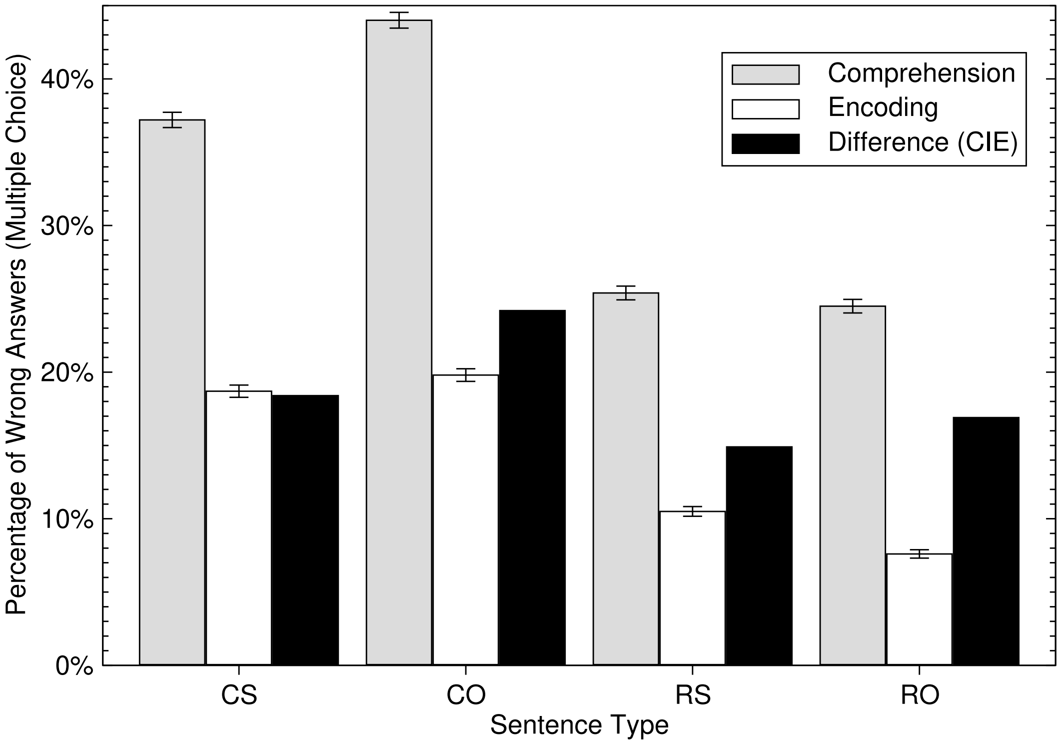


Figure 11.1: Question answering performance on the four RC sentence types in Experiment 11a. Gray bars show the error rate after comprehension, white bars show the error rate after just semantic encoding, and the black bars represent the subtraction, known as the *comprehension-induced error* (CIE).

subject-relatives over object-relatives. That is, the syntactic arrangement of object-relatives is more difficult than that of subject relatives.

Another way to test the amount of error introduced by syntactic processing is to measure the difference in the message layer representations resulting from semantic encoding and from comprehension. This difference, when measured with sum-squared error, is known as the *comprehended message error* (CME). When measured with CME, the results are only slightly different than with the CIE. The RS condition (12.03) is still easiest, but now the CS condition (13.78) is numerically, but not significantly, better than the RO condition (13.89). CO (15.17) is still the hardest. Therefore, in terms of their syntactic difficulty, it appears that the model finds the RS condition to be the easiest, the CO condition to be the hardest, and the CS and RO to be about the same. This suggests independent preferences for right-branching and subject-extraction. These results are in line with the relative frequencies of the structures in Penglish.

But why do we see such a strong semantic bias in favor of the right-branching conditions? As discussed in Section 6.2.4, the semantic system, due to its nature as a sequential auto-encoder, is expected to be sensitive to two main factors: frequency and confusability. We know that there is a frequency bias in favor of the right-branching conditions. However, the bias is not overwhelming and it does not explain the strong advantage for RO over CS in the question-answering performance of the semantic system.

An alternative explanation may be that there is a confusability difference between the center-embedded and right-branching conditions. What do we mean by *confusability*? The meaning of each of these sentences is encoded by four propositions, describing the subject and object of the matrix and embedded verbs, and the exact thematic roles played by them. About each sentence we could ask twelve different questions, by choosing a proposition and leaving out one of its three parts. Some of these questions will be easier or harder to answer based on the support or interference from similar propositions. In order to explain this more fully, we'll need some notation for labeling propositions. We'll refer to the matrix verb as M and to the embedded verb as E. The three nouns in a sentence will be labeled A, B, and C in the order in which they appear. A, of course, will always be the matrix subject. We will refer to a subject role as S and an object role as O.

A center-embedded subject-relative sentence has the following propositions: M-S-A, E-S-A, E-O-B, and M-O-C. The first proposition, M-S-A, says that the matrix subject is noun A. All four sentence types will share this proposition. E-S-A means that the embedded subject is also A, E-O-B means the embedded object is B, and M-O-C means the matrix object is C.[3] Certain questions will be easier to answer than others due to the consistency or inconsistency of these propositions. The question ?-S-A, for example, is very difficult in this case because there are two valid answers, M and E. The model has to make the correct choice based on the single bit that distinguishes the S relationships between the two clauses. However, the other questions about the first two propositions are easy. M-S-? and C-S-? are fairly easy to answer because the subject of a verb is always A. This may be true for the model as well as for human subjects who are asked fill-in-the-blank questions.

If one examines the difficulty of the 12 questions across all four sentence types, using any of a variety of models of interference, it turns out that the CS and RO sentences are semantically easier than the CO and RS. In the CS sentences, one noun, A plays the consistent role of subject, and in the RO sentences, B plays the consistent role of object. In CO and RS, one of the nouns plays both a subject role and an object role, resulting in an overall greater level of confusability.

This argument has certain similarities to the focus-switching account of RC processing (MacWhinney & Pléh, 1988). Both pertain to the problem of one noun being both a subject and an object. However, there is an important difference. Focus-switching assumes that the subject of the current clause plays a privileged role and that there is a cost associated with switching subjects over the course of a sentence. The confusability explanation I have put forth operates purely on semantic representations. It is not a theory of syntactic processing, but of memory.

Unfortunately, the confusability theory clearly does not explain the advantage for the right-branching sentences. It would only explain an interaction resulting in better performance on the CS and RO conditions relative to the others. In order to solve the right-branching preference mystery, it is necessary to look at the complete array of question-answering error rates. Table 11.3 shows the average error rates following semantic encoding on all 12 questions for each of the four sentence types, as well as averages for each row and column. The Proposition column indicates which

[3] Note that S and O, when used to label a proposition, are shorthand for a variety of relationships that could encode the subject or object. Depending on the verb, the subject may be an agent or experiencer and the object may be a theme or a patient. These fields also may contain an emphasis bit indicating the subject or a bit indicating that the relationship spans an RC boundary.

Type	Proposition	Left	Middle	Right	Average
CS	M-S-A	6.0	9.3	1.0	5.4
	E-S-A	18.3	28.5	5.6	17.5
	E-O-B	51.9	16.5	50.0	39.5
	M-O-C	14.9	5.0	17.8	12.6
	Average	22.8	14.8	18.6	18.7
CO	M-S-A	5.8	36.9	4.0	15.6
	E-O-A	17.9	36.0	12.8	22.2
	E-S-B	26.9	38.8	32.1	32.6
	M-O-C	9.3	2.4	14.7	8.8
	Average	15.0	28.5	15.9	19.8
RS	M-S-A	4.9	6.4	4.6	5.3
	M-O-B	11.0	9.6	15.6	12.1
	E-S-B	2.8	26.9	7.8	12.5
	E-O-C	17.8	8.2	10.1	12.0
	Average	9.1	12.8	9.5	10.5
RO	M-S-A	5.7	4.6	3.6	4.6
	M-O-B	6.5	6.5	3.5	5.5
	E-O-B	2.9	31.4	4.6	13.0
	E-S-C	4.6	12.4	4.6	7.2
	Average	4.9	13.7	4.1	7.6

Table 11.3: Error rates on each of the possible questions following semantic encoding in Experiment 11a.

of the four propositions for the sentence is being queried. The Left column is the error rate when the model is asked to provide the left element, or action (verb), of that proposition. The Middle column is the error rate for providing the relationship between the action and object and the Right column is the error for providing the object (noun). For each sentence type, the order of the propositions, from top to bottom, is the order in which they were fed into the semantic encoder. For the case of CS, the propositions were loaded in the order: M-S-A, E-S-A, E-O-B, M-O-C.

With the exception of the CO condition, the model is quite good at answering questions about the first, M-S-A, proposition. Note, however, that the order in which the propositions for the two clauses are loaded into the encoder differs between the center-embedded and right-branching conditions. In the right-branching conditions, the propositions for the matrix clause are loaded first, followed by the propositions for the embedded clause. This is the natural order because a large proportion of sentences in Penglish have a single transitive clause, in which the subject would be loaded first and then the object. The semantic encoder, which is essentially trying to compress these propositions into a single, finite representation, probably learns to take advantage of the typical clause order. The subject and object proposition for a single clause may be bound together in some way by the encoder. The model need not represent the verb independently, because it is usually the same for these two propositions. When an object proposition comes in right after a subject, the encoder will be adapted to the case in which the verb is the same.

However, because of the way I ordered the propositions in relative clause sentences, center-embedded RCs are different. The two propositions for the embedded clause intervene between the subject and object propositions for the matrix clause. It is likely that this intervention causes encoding problems which are the main reason for the center-embedding disadvantage that we see. Error rates in the center-embedded conditions are higher than in their right-branching counterparts for both embedded clause propositions, but more so for the second one of the two. Error rates are also especially high in answering questions about the verb, or left part, of the embedded clause propositions in the center-embedded conditions. This suggests that there is interference and loss of information when the model's hypothesized expectation that it will receive two propositions in a row using the same verb is violated. Thus, the model's strange performance may largely be due to a poorly chosen method of ordering propositions for semantic encoding.

Why was the previous version of the model, which had a preference for center-embedded sentences, not subject to the same problems? That model used a very different method for encoding the propositional structure of a sentence.

A simple transitive relationship in the current model is encoded with two propositions, one for the subject and one for the object. In the previous model, however, this was encoded with a single proposition in which the verb, or action, formed the relationship between the subject and object. As a result, there was just one proposition per clause and the subject and object could not be separated by intervening propositions. This method of propositional encoding was abandoned in favor of the new one because it was not rich enough to express the diversity of thematic roles used in Penglish.

Although the CSCP model does not show the appropriate order of preference in its comprehension performance across the RC sentence types, this experiment has revealed some interesting properties of the model, as well as a problem with its design. A variety of factors are at play in the processing of these four relatively simple sentence types. Frequency biases favor the subject-relative and right-branching conditions, but mainly work against the CO condition. When syntactic difficulty is distinguished from semantic difficulty, there seems to be a moderate syntactic bias in favor of the subject-relative conditions as well as a weaker one in favor of the right-branching conditions. The subject-relatives are presumably easier because they use the typical SVO word order. Center-embeddings may cause problems because they interrupt the main clause and force the model to integrate the subject and main verb across this gap.

However, there are also important semantic factors at work in the model. In theory, the confusability suggests that the harder conditions to encode semantically should be the CO and RS because of the dual roles played by one of the nouns. But there seems to be a much stronger bias in favor of easier semantic encoding in the right-branching sentences. A likely explanation for this is that it results from the way in which propositions were ordered for the semantic encoder. If this bias were eliminated, it is likely that the model would show much improved performance in the center-embedded conditions. Nevertheless, it is still probable that CO would remain the hardest condition.

Reading times

Our discussion of Experiment 11a would not be complete without also looking at reading times of the RC sentences. Figures 11.2 and 11.3 show the word-by-word reading times on the center-embedded and right-branching sentences, respectively. Because the conditions are matched, reading times are identical up to the relative clause. It is not very enlightening to directly compare reading times on the RC between the CS and CO word conditions because the word order differs. However, reading time on the word immediately following the RC, which is the main verb in the center-embedded case and the end-of-sentence in the right-branching case, is significantly longer following the object-relatives. This indicates that the model has more difficulty recovering from an object- than from a subject-relative.

Figure 11.4 compares reading times in the four conditions on just their relative clauses. *That* is read more slowly for the center-embedded RCs. This is probably a frequency effect, as post-modification is more common after matrix objects. The word following *that*, which is either a verb or an article, is read quite slowly. Of course, the articles in the object-relatives are read faster than the verbs in the subject-relatives, but the former are actually fairly slow as articles go.

Reading time on the NP within the RC (RC1–RC2 for subject-relative, RC2–RC3 for object-relative) is roughly equal across the conditions. However, reading time on the verb is faster for the object-relatives. As a result, the overall reading times are faster on the object-relatives than on the subject-relatives. This does not match the results of Gibson et al. (in press). One possible explanation is that this is a result of interference from sentential complements. The relative frequencies of different types of RCs were carefully controlled in the Penglish language. So too were the frequencies with which individual verbs occurred with sentential complements. However, the relative frequencies of sentential complements and relative clauses was not controlled. It turns out that, in Penglish, sentential complements occur 2.2 times more often than relative clauses. However, in English, RCs actually occur 1.3–1.6 times more often than SCs, depending on how one counts.

The reason this is an issue is that the word order in a sentential complement is the same as the word order in an object-relative, including the fact that they both tend to start with *that*. Therefore, the model is very used to dealing with clauses having this word order, which may lead to faster word reading times. That there may be such a confusion would account for the fact that reading times following object-relatives are slower than those following subject-relatives. The model may be expecting another NP at that point, as would occur in a sentential complement. This would suggest that, if the frequency of sentential complements in Penglish were reduced to better match English, reading times on object-relatives would increase. Incidentally, the previous version of the model, which had a strong

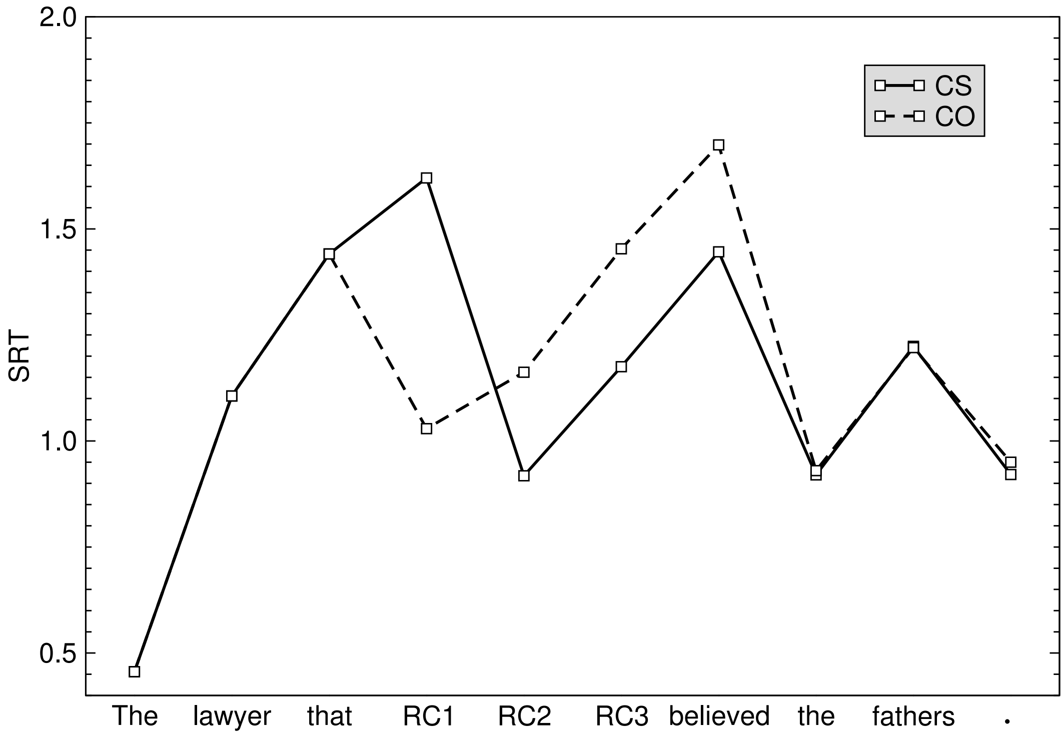


Figure 11.2: Reading times on the center-embedded sentences in Experiment 11a.

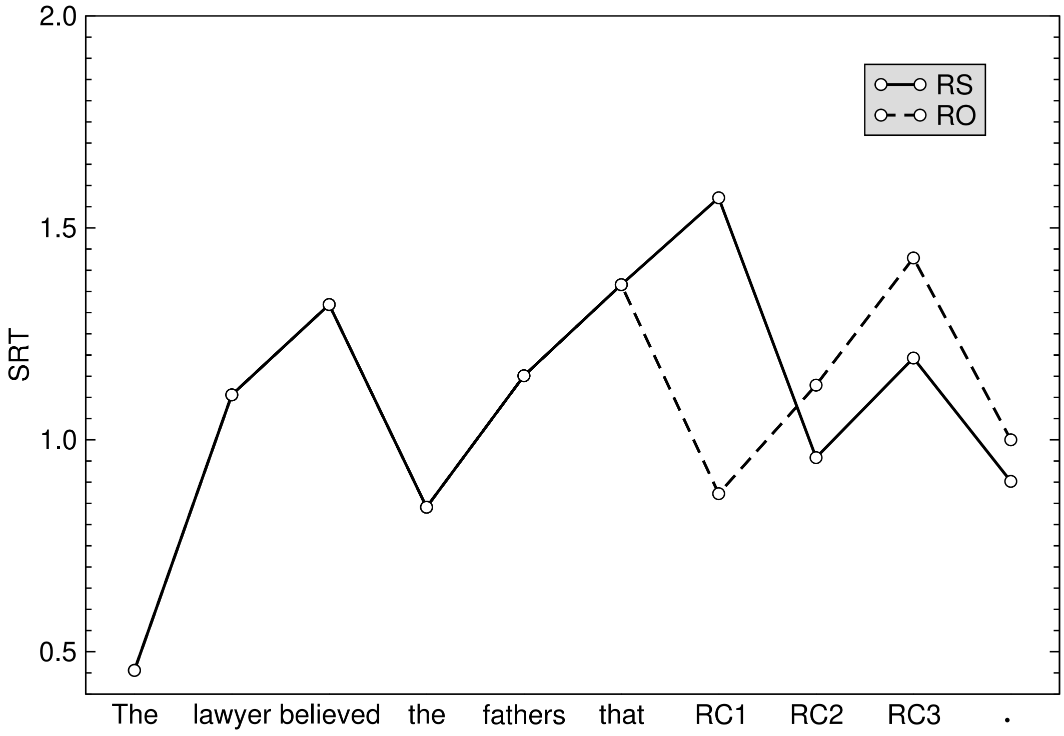


Figure 11.3: Reading times on the right-branching sentences in Experiment 11a.

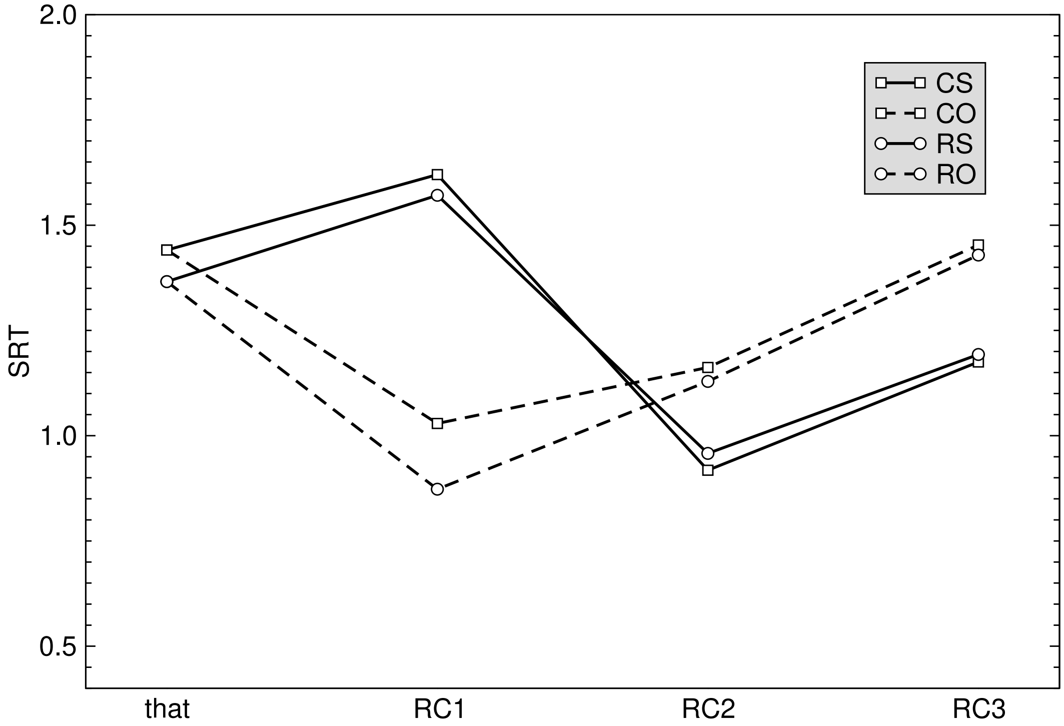


Figure 11.4: Reading times on the relative clauses in Experiment 11a.

preference for subject-relatives, was never exposed to sentential complements.

11.2.2 Experiment 11b

Although the model clearly has some problems that need to be worked out, we can still look at other issues in the processing of RCs that may be orthogonal to the relative difficulties of the four main single RC sentence types. In the materials used in Experiment 11a, the semantic relationships within the sentences were fully counterbalanced, making the sentences maximally confusing. In terms of structure, the sentences used in Experiment 11b were similar to those in Experiment 11a except that NPs were not required to have articles (if they were plural, for example) and the verbs were allowed to take any active transitive form. However, the new items were not generated from a template but were extracted from a corpus of Penglish using TGREP2. One hundred sentences of each type were selected. The question is, will the model's performance improve significantly if more natural relative-clause sentences are used and what will be the effect on the relative difficulty of the four RC sentence types?

Figure 11.5 shows the comprehension measures corresponding to those for Experiment 11a, which were given in Figure 11.1. First note that, overall, the comprehension performance has improved significantly and that the message encoding performance has improved even more impressively. This is particularly true of the right-branching sentences. The average center-embedded comprehension error rate dropped from 40.6% to 24.8% while the average right-branching comprehension error rate dropped from 25.0% to 4.0%. Error rates for message encoding dropped from 19.3% to 4.4% for center-embeddings and from 9.1% to 0.54% for right-branchings.

We can draw several conclusions from this. The model is definitely sensitive to the frequency and plausibility of the semantic relationships in RC sentences. Despite the fact that the verb phrases were more complex in the items in this experiment, using imperfect, progressive, and present and future forms, error rates were significantly lower. The more natural items did not have much effect on the subject-relative versus object-relative difference, but there was a large effect on the center-embedded versus right-branching difference. Error rates in the center-embedded conditions, for both encoding and comprehension are now far above those for the right-branching conditions. This provides further support for the theory that the unusual proposition order in the center-embedded conditions may be the major

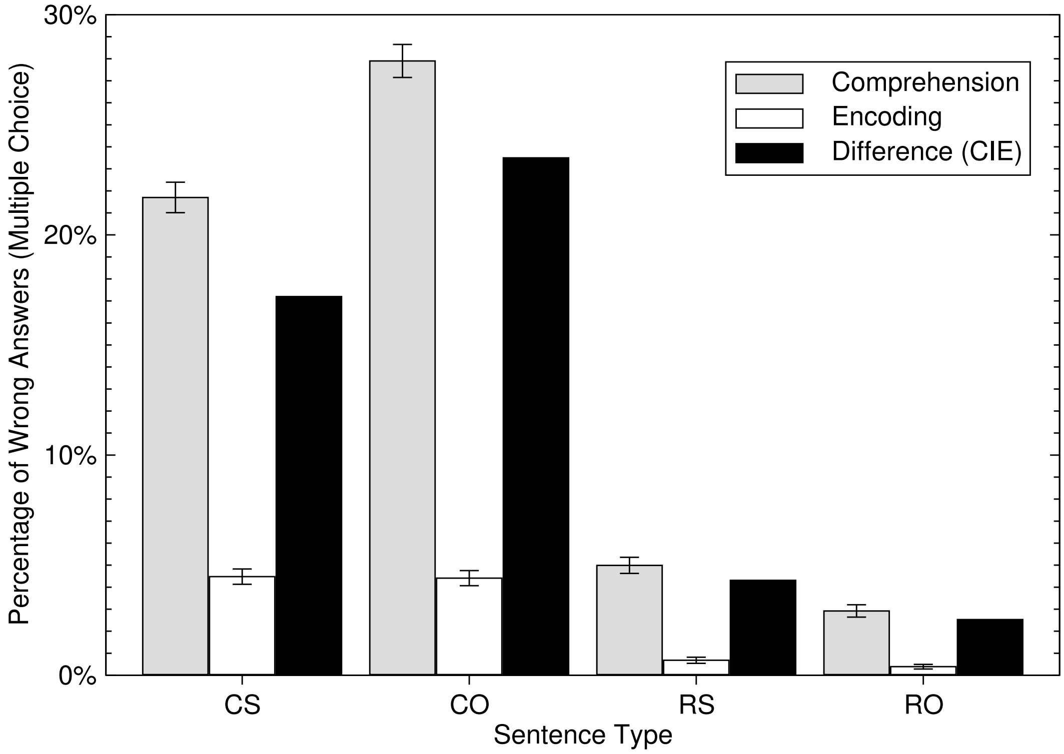


Figure 11.5: Question answering performance on the four RC sentence types in Experiment 11b. Gray bars show the error rate after comprehension, white bars show the error rate after just semantic encoding, and the black bars represent the subtraction, known as the *comprehension-induced error* (CIE).

contributor to the right-branching bias in the current model.

11.3 Discussion

Unlike its predecessor, the current version of the CSCP model does not replicate the fairly consistent empirical findings on the comprehension and reading-times differences between various single relative-clause sentences. Although subjects seem to perform better on center-embedded and subject-relative clauses, the model now has a strong preference for right-branching structures, with right-branching object-relatives being the most-favored condition and center-embedded object-relatives being the least favored.

I have argued that this is due to two main problems with the model. One is that the order in which propositions are loaded into the semantic encoder is non-standard in the center-embedded conditions. Although the semantic representations are meant to be independent of syntax, the current method of ordering the propositions is actually dependent on syntax. The strong effect this would have on the model was not anticipated. In the future, propositions should be fed into the semantic encoder either in random order or in an order that groups together any propositions pertaining to a single clause or phrase. The former would make the semantic representations less sensitive to syntax, but it would tend to de-emphasize the relationship between the matrix subject and verb, which, arguably, is the most important in the sentence. The latter method would probably make the encoding task easier for the model because related propositions could more easily be grouped together, but it may not sufficiently maintain the intended semantic/syntactic distinction.

The second major problem is not in the model itself but in the Penglish language, which uses too many sentential complements in proportion to the number of relative clauses. In order to perform the intended experiments on the sentential complement ambiguity, a large number of SC verbs were included in the lexicon. In order to properly balance the proportion of RCs and SCs, it may be necessary to eliminate some of these SC verbs or add some more non-SC verbs. While increasing the size of the lexicon is important in advancing the model, it may mean that a larger network and/or more training will be necessary to achieve comparable performance levels, possibly requiring faster computers than we are currently using.

Some additional experiments on RCs were planned but were not performed because it did not seem worthwhile to pursue them given the aforementioned problems. One is the finding of Gibson and Warren (1998) that single and self-embedded object-relatives are deemed to be less complex when they contain first- or second-person pronouns. One explanation for this finding is that the referents of these pronouns are already in the current discourse and are thus accessed more easily. An alternative hypothesis is that this is simply an effect of the short length and high frequency of these pronouns. If the model, which lacks a discourse representation, shows similar effects, it would be suggestive that they can be accounted for simply by frequency and possibly length. Additionally, it would be interesting to test the model on the grammaticality decision results obtained by Gibson and Thomas (1997) on nested RC sentences.

In searching for explanations for the strange behavior of the model in processing relative clauses, a lot of consideration was given to the range of factors that may affect the semantic difficulty of the various relative-clause types. Each of the twelve possible questions that could be asked of the network was more or less difficult across the conditions. This lead to the thought that perhaps previous empirical studies did not go far enough in evaluating the comprehensibility of such sentences. If only a subset of the relationships in a sentence were ever queried, the measured difficulties may not be representative of the comprehensibility of the sentence as a whole. Thus, it is important that all parts of the meaning of the sentences be tested over the course of an experiment.

Another important issue is the word order used in the questions. If the word order of a question matches that of the corresponding part of the sentence, it will probably be easier for subjects to answer the question because they can do so by recalling the word order of the sentence and filling in the missing part or detecting differences. There are two problems here. If all questions are asked in an active form, which is typical, answering questions will likely be easier in subject-relative sentences, in which all clauses use the standard active word order. This may create an artificial bias in favor of subject-relative conditions. But more generally, if subjects are able to answer questions by recalling the sentence from a phonological buffer and reprocessing it, question-answering performance will not accurately reflect first-pass processing. This is fine if one is not interested in first-pass sentence processing, but few researchers in the field seem to be interested in our ability to comprehend a sentence after three or four tries.

To provide a better point of comparison for the model, I have begun running an experiment with human subjects that attempts to eliminate some of the problems of previous RC studies. The same four sentence types are being used, but the items are fully counterbalanced, as in Experiment 11a. Across subjects, each item appears in all four different sentence types and in all twelve possible arrangements, for a total of 48 conditions. This should remove any semantic biases across the sentence types, other than those due to confusability between clauses. Questions are fill-in-the-blank with four possible responses, rather than five as in the model. Two questions are asked about each sentence, one pertaining to the matrix clause and one to the embedded clause. The questions require subjects to fill in either a verb or a noun and can be phrased in either active or passive voice. The voicing of the questions is expected to interact with whether RCs are subject- or object-relative.

The experiment as just described will still suffer from the problem that subjects may be answering comprehension questions not based upon what they extracted from the first reading, but by internally replaying and reparsing part of the sentence from a phonological memory buffer. If that is true, the results may not reflect the ease of initial analysis, but the ease of memorization and recall, which could differ significantly between center-embedded and right-branching sentences. Therefore, a second version of the experiment is being run in which we attempt to interfere with any use of phonological memory to prevent this post-analysis. After the sentence is read and before the questions are given, three syllables are shown on the screen. Subjects are asked to say the syllables repeatedly, out loud and in rhythm, over the course of the question answering. It is hoped that this will impair subjects' ability to revisit the surface form of the sentence and force them to rely on any meaning extracted in the first pass. The results of the two versions of this experiment should broaden our understanding of how we comprehend relative-clause sentences and should also provide a more appropriate standard by which to assess the CSCP model.

Chapter 12

Production

Historically, language production has received less attention than language comprehension. There are a variety of reasons for this. Traditional linguistics has generally focused on surface forms and the deep syntactic structures hypothesized to underlie them, with little emphasis on semantic representation or processing. With semantics out of the equation, the only major sentence-level process of any interest is parsing, or the mapping from surface form to deep structure. The reverse mapping, from deep structure to surface form is generally of little interest because it is fairly trivial. Given a context-free representation of a parse tree, producing a sentence is simply a matter of walking the tree in the right order and finding the leaves. The only interesting issues that arise in production may be those regarding transformations.

However, once meaning is added as a critical representational layer, things change considerably. True production, or mapping from semantics to surface form, is not at all trivial. Arguably, between production and comprehension, production is the more difficult task. Most second-language learners would probably agree. As Bock (1995) points out, a comprehender can often safely ignore redundant aspects of syntax, such as a agreement or gender markings. A producer, on the other hand, must get everything right.

Language production certainly deserves as much interest and respect as comprehension. But even today there seems to be much less research done on sentence production than there is on comprehension. The reason for this is largely practical. Experiments on production are simply harder than those on comprehension. Because production is by nature open-ended, and because semantic representations are essentially hidden, it is hard to elicit particular sentences or types of sentences. Researchers who wish to study the production of a specific structure must be especially creative in their experimental design. The other problem with studying production is that analyzing the output is very time consuming. Reading time or comprehension results can be extracted and analyzed on a computer with a few simple commands. But the output of a production experiment must generally be hand-coded by a trained linguist (i.e., graduate student), and the phenomena of interest, such as speech errors, are often quite rare.

Production experiments are also more difficult than comprehension experiments to conduct on the CSCP model, although the model is easier to work with than are human subjects. Because its semantic representations are accessible, it is at least possible to request that the model produce a particular sentence. Furthermore, the model produces its output in text rather than spoken form. But analyzing the model's outputs in any automated fashion still presents a problem, particularly in cases where they are not grammatical. So analysis must often be done with a mixture of customized parsing procedures and good old-fashioned human effort. All this is to say that, in studying the behavior of the CSCP model, production has been treated less extensively than comprehension. However, a few interesting results have been obtained and they are presented here.

Section 12.1 discusses word-by-word production and analyzes the model's performance using a variety of measures and on various sentence types. Section 12.2 considers free production and discusses the sorts of production errors commonly made by the model. One particular class of errors that are committed fairly often by speakers of English are illustrated in this sentence by the incorrect agreement of the plural verb *are* with the singular subject *class*. When an NP that differs from the subject in number intercedes between it and the main verb, speakers and writers are often mislead into matching the number of the verb to that of the second NP. As shown in Section 12.3, the CSCP model shares this tendency. Finally, Section 12.4 explores the phenomenon of structural priming.

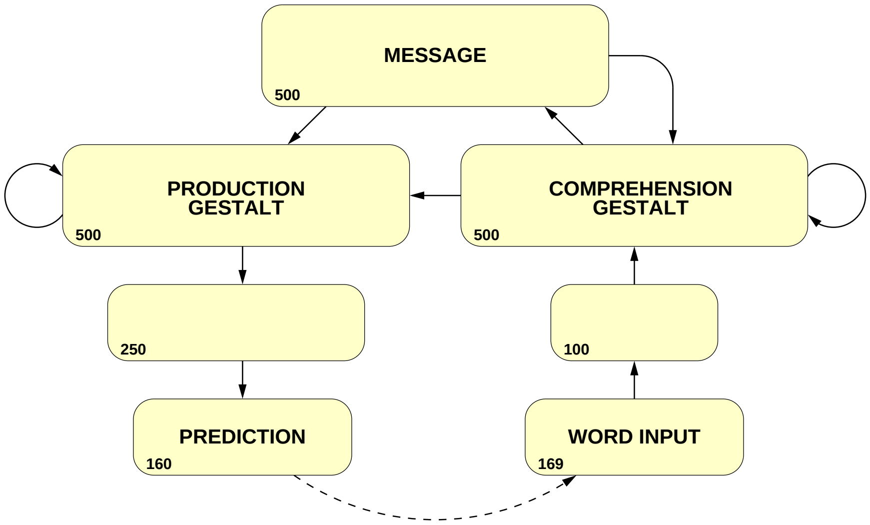


Figure 12.1: The comprehension, prediction, and production system.

12.0.1 Simulating production

Section 6.3.2 discussed how production is simulated in the CSCP model; the following is a brief review. For convenience, Figure 6.4, depicting the comprehension and production system, is reproduced here as Figure 12.1. There are actually two forms of production in the model: word-by-word and free production.

Word-by-word production is identical to comprehension, except that the entire message is known in advance and the activations of the message layer units remain hard clamped to the encoded message throughout the sentence. Once these activations have been set, the model is given the start symbol at the word input layer and activation flows through the network, leading it to predict, or produce, the first word of the sentence at the prediction layer. This layer is actually composed of two parts, representing the word's stem and ending, each part using a localist encoding. The endings include the most common inflectional morphemes, such as -d, -ing, -ly, -n, -o, -z, and nothing. The most active stem unit and the most active ending unit determine the word that would be produced by the model. For example, the stem eat plus the ending -ing would form the word *eating*. Hopefully the correct word is the most active one, but when it is not we can get a sense of how wrong the model was by measuring the rank order of the unit activation representing the stem or ending of the word. In word-by-word production, the word produced by the model is tested, but not actually used. Instead, the correct next word in the sentence is fed into the network and the process is repeated.

In free production, on the other hand, each of the model's productions is actually used as the next word in the sentence. The model repeatedly produces a word and then "listens" to the word it just produced, leading to the production of the next word. This continues until the model produces the end-of-sentence symbol. If the model gets confused and begins babbling endlessly, its output is capped at 25 words, but this limit is rarely if ever reached. Free production is identical to word-by-word production until the model makes a mistake. At that point, an interesting question is what happens next. The networks have never actually been trained on ungrammatical or otherwise incorrect input. Will production break down with the first error and degenerate into nonsense, or will the model be able to recover from a mistake and continue on with the sentence?

Clearly, comprehension and production are largely integrated in the CSCP model. Both of them use the same groups of units and the same links between them. But it would not be exactly fair to say that comprehension and production are equal partners. Comprehension could proceed without the production end of the system. Although prediction is not a necessary component of comprehension and the networks' predictions are not actually used during comprehension, it is expected that prediction would be very useful in more complete instantiations of the model. In a truly fully recurrent model, accurate predictions should allow the model to anticipate upcoming words, reducing the time required to process them. In a system that must actually recognize words from visual or auditory input, prediction may make that portion of the system faster to respond and more tolerant to noise. But prediction is not strictly necessary for comprehension, and comprehension should be relatively successful, if somewhat delayed, even

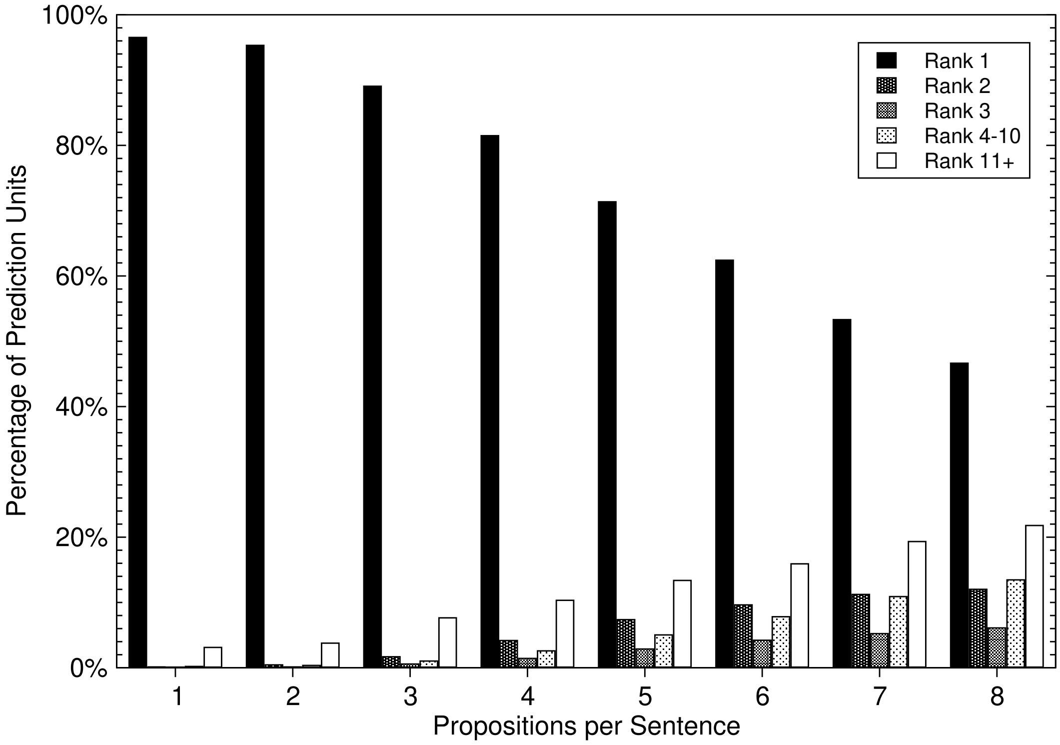


Figure 12.2: Rank, in terms of activation level, of the correct word stem units. Rank 1 means the correct unit was the most active one, while rank 2 means it was the second most active.

when predictions are inaccurate.

On the other hand, the comprehension system in this model is necessary for production. Production cannot proceed in a closed loop, except possibly for repetition of memorized surface forms, but relies on internal feedback through the comprehension system. This leads to the prediction that it should not be possible to have fully intact production with impaired comprehension, as long as the impairment to comprehension occurs between the lexical and message representation, and not at higher semantic levels. It should be noted that others working in the area of connectionist language processing, including Plaut and Kello (1999), Christiansen and Chater (1999b), Chang et al. (2000), have also proposed that comprehension and production are closely related, although not necessarily in exactly the manner employed in the current model.

12.1 Word-by-word production

We begin with some general measures of production ability, in both its word-by-word and free production guises, starting with word-by-word production errors as a function of the number of propositions per sentence. As mentioned in the previous section, word-by-word production can be measured by examining the rank of the correct word among the model's predictions. If the unit of the prediction layer corresponding to the correct word has the highest activation overall, the word will have rank 1. If the correct unit has the second-highest activation, the word will have rank 2. Actually, each word will have two ranks, one for its stem and one for its ending. A word is only produced completely correctly if both its stem and ending have rank 1.

Figure 12.2 shows the distribution of word stem production ranks that occur for sentences with from one to eight propositions. For 1-prop sentences, the correct word stem has rank 1 96.5% of the time. It has rank 2 only 0.11% of the time and rank 3 0.05% of the time. Thus, the incorrect stem is produced in just 3.5% of the cases. When an error is made, the rank of the correct stem is almost always not even in the top 10. Thus, when the model is wrong, it tends to be very wrong. For 4-prop sentences, 81.5% of word stem predictions have rank 1 and 4.17% have rank 2. Thus,

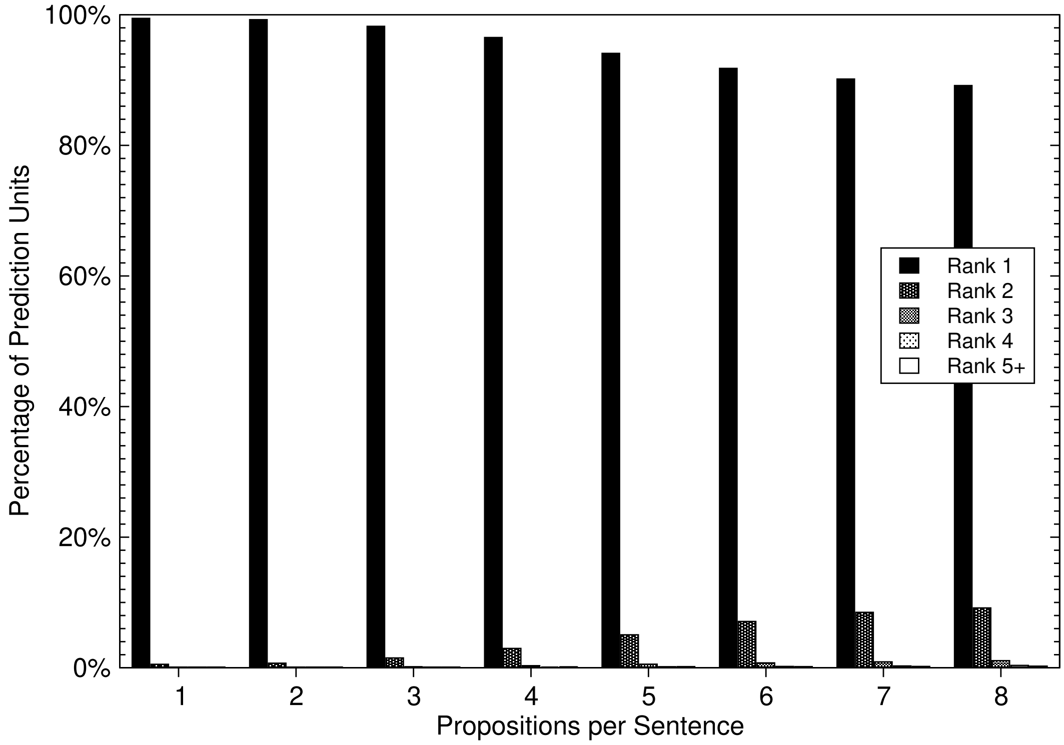


Figure 12.3: Rank, in terms of activation level, of the correct word ending units. Rank 1 means the correct unit was the most active one, while rank 2 means it was the second most active.

with more complex sentences, there is a greater tendency for the model to be close to correct if there is an error.

Figure 12.3 shows similar prediction rankings for the word endings. In general, predicting the word ending is much easier, largely because there are 153 stems in Penglish, but just 7 endings, so there are more opportunities for error in producing the stem. For 4-prop sentences, the model still gets 96.5% of the endings correct. When it does make a mistake, the correct ending is usually the second most active one. Even in 8-prop sentences, the correct ending is in the top two in over 98% of the cases.

A word production is wrong if either the correct stem or ending fail to be rank 1. If there is an error in either the stem or ending prediction, it will be called a *word* prediction error. We might ask whether the errors made in predicting the stem and ending of a word are independent, or if they tend to co-occur. If the errors are independent, it would suggest that the stem and ending are produced by two functionally distinct mechanisms. But if they are highly correlated, it would suggest a unified mechanism. One way to test the degree of independence of the stem and ending prediction errors is to compare the combined rate of word errors, the cases in which either the stem or ending is wrong, with the rate that would be expected if the two predictions are independent.

Figure 12.4 shows the error rates for the stem and ending predictions as a function of the propositions per sentence, as well as the actual word error rate and the word error rate that would be expected if independence were true. If the actual word error rate were higher than the independent rate, it would indicate that the stem and ending are anti-correlated. But because the actual word error rate is less than the independent error rate, it suggests that the stem and ending predictions are, in fact, correlated. The word error rate is in fact only slightly higher than the stem error rate, suggesting that few word production errors are due to a mistake on just the ending. Across the testing corpus, the frequency of ending errors was 5.26%, and the frequency of stem errors was 26.27%. If these were independent, the frequency of word errors would be 29.20% $(1 - (1 - 0.0526)(1 - 0.2527))$, but the actual word error rate is only 27.42%.

Another way to test the degree of correlation is to compare the rate of stem errors in cases when there is or is not an error predicting the ending. If the stem and ending errors are independent, these rates should be the same. Across the complete Penglish testing corpus, the overall frequency of stem errors was 26.3%. But the rate of stem errors in cases

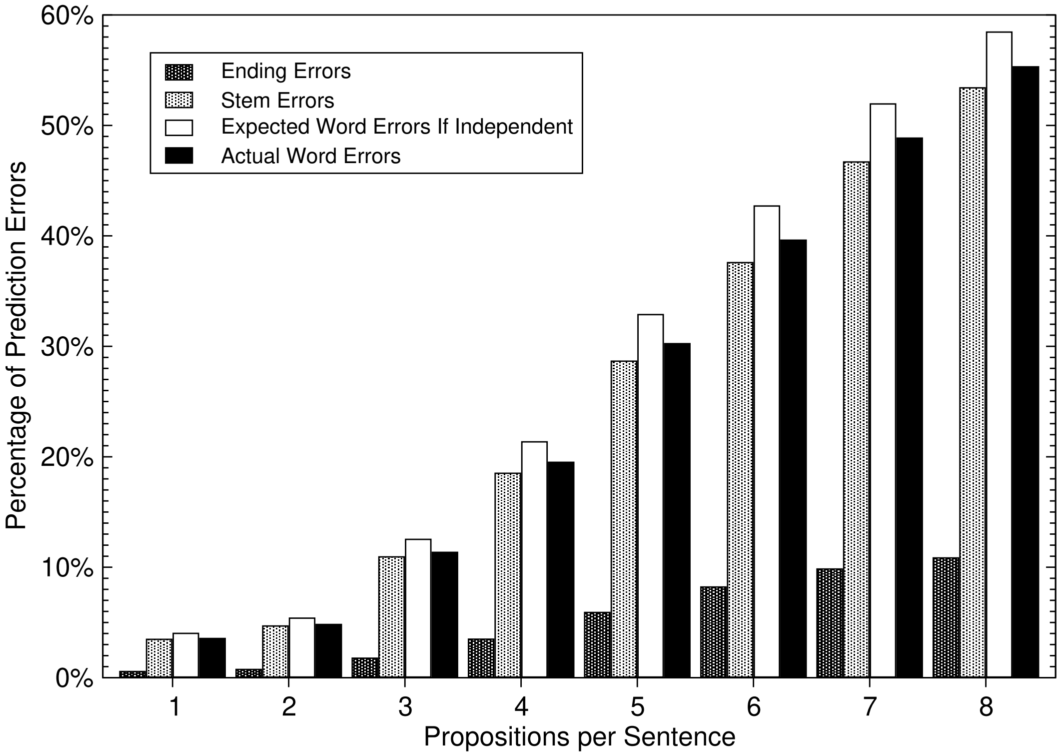


Figure 12.4: Prediction error rates as a function of the number of propositions per sentence. "Actual Word Errors" are cases in which there was an error on either the stem or the ending. "Expected Word Error If Independent" is the expected frequency of word errors if stem and ending errors were independent of one another.

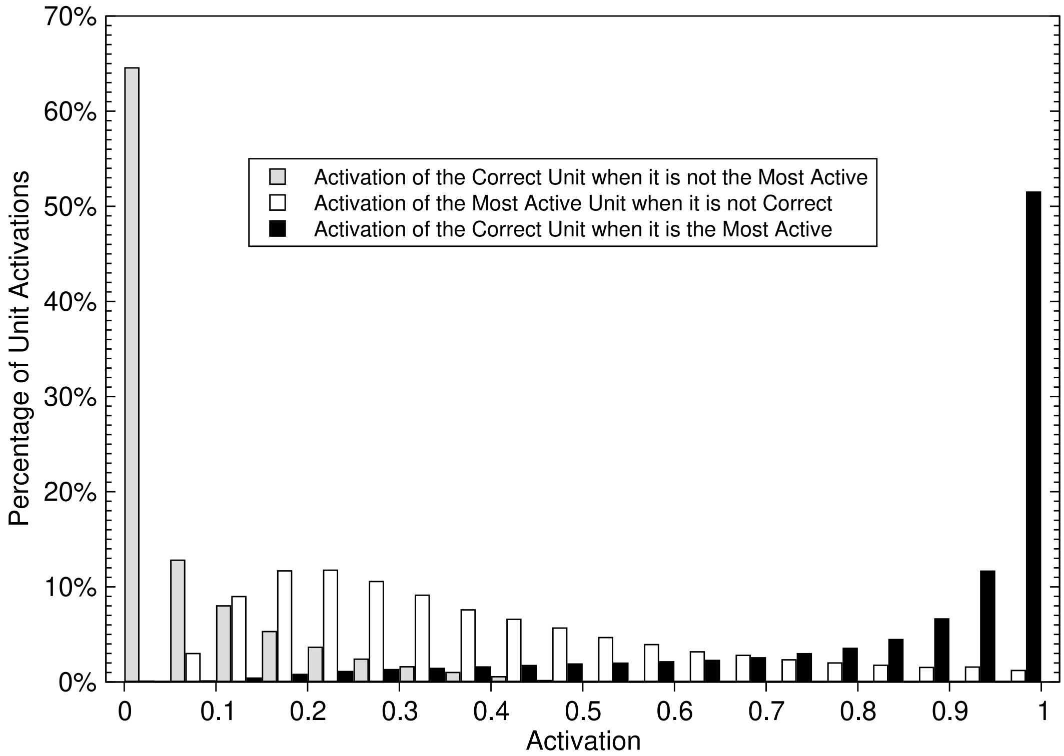


Figure 12.5: Three histograms showing the distribution of activations for word stem units in different circumstances.

with no ending error was 23.4%, while the rate of stem errors in cases with an ending error was 78.1%. Therefore, when a prediction error occurs on the word ending, it is much more likely to occur on the word stem as well.

These results are merely suggestive of a single mechanism for predicting word stems and endings in the model. There are, of course, reasons to expect that stem and ending errors would be correlated even if they were produced by completely independent mechanisms. Sentences that are difficult will probably tend to cause more of both types of errors, resulting in a correlation regardless of the type of system responsible. We can reduce such biases by looking at error rates on particular sentence types, for which there will be much less variance in difficulty between sentences. On simple transitive sentences, the stem error rate was 4.2% in cases with no ending error, but 72.7% in cases with an ending error. Thus, the correlation was actually stronger. Similarly, on sentential complement sentences, the stem error rates were 8.6% and 67.1%. Therefore, even when controlling for sentence type, there is a very strong tendency for prediction errors in the word stem and ending to co-occur.

Another interesting question is how committed the model is to its predictions. Is the model highly committed to predictions when they are correct, or is the unit with highest activation only slightly more active than its competitors? When the model makes a mistake, is it strongly committed to that mistake or is it unsure of its answer, and is there a clear difference in strength between correct and incorrect predictions? Figure 12.5 shows three histograms of the activations of stem prediction units in various circumstances. The black bars depict the distribution of activations of the correct stem unit when it is also the most highly active one. Because the distribution is so skewed to the right, it indicates that, when the model is correct in its prediction, it tends to be very confident in that prediction. In contrast, the white bars show the activation of the winning, or most active unit, when that unit is not the correct prediction. If the model were as confident in its wrong predictions as it is in its correct predictions, the white and black distributions would be similar. It is clear that they are not and that the white bars are skewed much more to the left. Thus, when the model is wrong, it is not as confident in its predictions as it is when it is right. Finally, the gray bars show the distribution of activations of the correct unit when that is not the model's strongest prediction. This distribution is highly skewed to the left, indicating that when the model is wrong, the correct prediction is rarely in close competition with the winning unit. It is not exactly clear what to make of these findings, but they seem to provide another little window into the model's behavior.

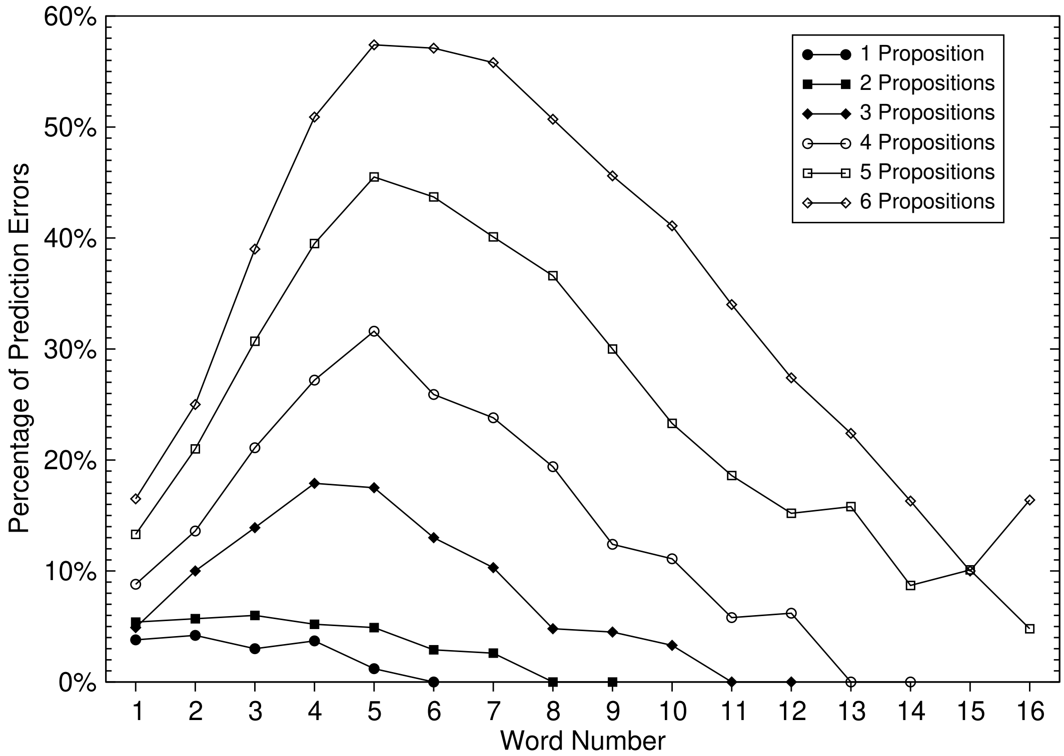


Figure 12.6: Frequency of word prediction errors as a function of the number of propositions and the position of the word in the sentence.

Finally, we consider the difficulty of word predictions as a function of the position of the word in the sentence. A valid expectation might be that predictions will tend to become worse over the course of the sentence. However, as shown in Figure 12.6, that is clearly not the case. The word prediction error rate is shown as a function of the position of the word in the sentence for items with from 1 to 6 propositions. Predictions for simple sentences, with 1 or 2 propositions, actually tend to become better over the course of the sentence, while those for complex sentences have a pronounced inverted-U shape. There may be several reasons for this. A simple one is that the model is quite good at predicting the end of the sentence, when it is appropriate to do so, which will reduce the error rate for the higher word numbers. However, the error rate for the complex sentences peaks on the fifth word and then begins to decline. This cannot be explained by the end-of-sentence factor because almost none of these sentences are ending by the 6th word.

An alternative explanation for the U-shaped difficulty of word-by-word predictions over the course of a sentence is the following. The first few words are relatively easy because most sentences start with the main subject NP. Then things quickly become more difficult as embeddings and verbs must be produced. In particular, there is the problem of ordering all of the constituents of the sentence. Not only must the model produce each constituent correctly, it must produce them at the correct time. Towards the end of the sentence, many of the constituents have already appeared, and the model seems to be fairly good at making use of that fact. It can then focus on producing the few constituents that remain, which is an easier task. This account seems to be supported by the investigation of individual free production errors in Section 12.2.1.

12.1.1 Word-by-word production by sentence type

Word-by-word production was tested on 27 different sentence types and the resulting error rates are shown in Figure 12.7. Included are the 25 sentence types described in Table 7.1 in Chapter 7, as well as the two basic right-branching RC types RC-RS and RC-RO. Error rates on most of the simple sentence types are relatively low. The model makes fewer than 6% errors on the INTR, PASS, INTR-ADVB, INTR-VPP, TRNS, PASS-BY, TRNS-ONPP, and TRNS-VPP types. The model is quite good at producing PPs. This is especially true for TRNS-ONPP, in which a single PP mod-

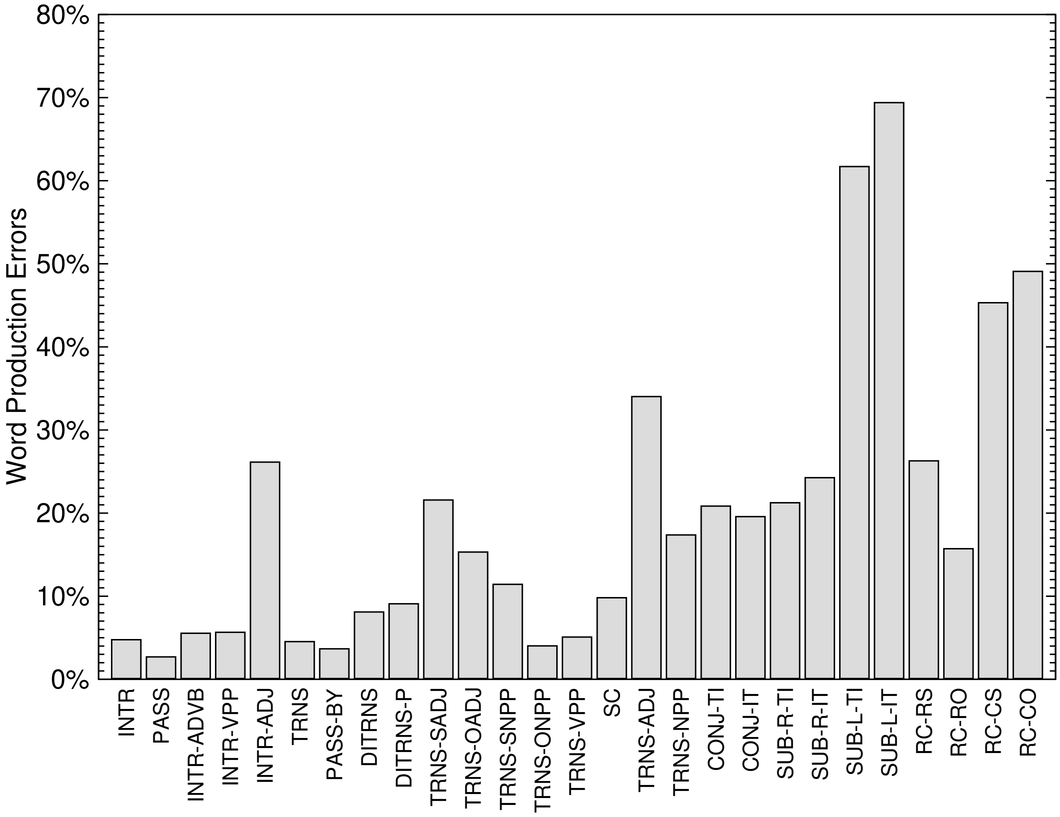


Figure 12.7: Percentage of word-by-word production errors for a variety of sentence types.

ifies the direct object, and for TRNS-VPP, which ends in a verb-modifying PP. The model is noticeably worse in the TRNS-SNPP case, in which the PP modifies the subject. This could be due to the fact that PPs modifying the subject are less than half as common as PPs modifying the object. But it also may have to do with the difficulty of an embedding preceding the matrix verb.

Interestingly, the model is very good at producing passives—better, in fact, than it is at actives. This does not accord with what we know about human performance. It seems likely that the reason for this has to do mainly with the method used to encode the propositional representation of a sentence's meaning. If one were using a single static thematic role frame for a complete action, one would normally have a slot for the agent, one for the verb, and one for the patient. Because actives are more common, the model would become used to associating the agent with the subject position and the patient with the object position. Passives would be more difficult because they require a reversal of this normal mapping.

However, the current model uses separate propositions to encode the various thematic roles, and these propositions are fed into the encoder network one-by-one to produce a complete message. The order in which the propositions are encoded is potentially flexible, but the current scheme uses a fixed order, which is always subject first. The reason for this was that the subject is thought to be the most important, or stressed, element. In an active transitive, the subject/agent is loaded first, followed by the patient/object. In a passive transitive, the subject/patient is loaded first, followed by the agent/object. If the model were to rely on the thematic role designations of the actives to determine the order of the NPs during production, it would have trouble with passives. But, to the model, the more salient cue seems to be the order of the propositions. The first proposition, whether active or passive, is always the subject and that should be produced first. Therefore, in this encoding scheme, there is no significant hindrance to comprehending or producing the passive construction.

One might still ask why passives seem to be even easier than actives when it comes to production. The reason for this may have to do with the verb structure, as most of the model's errors in producing simple sentences seem to occur on the verb. Passives tend to have more complex verb structures, with one or more auxiliaries. Apparently, these do not cause as much trouble for the network as one might expect. Perhaps this is because the inflected part of the verb, which is always some form of the verb *to be*, is separated from the main verb. Thus, the model can concentrate on producing the inflection first, using the common auxiliary verb, and then on producing the more difficult main verb. Perhaps the separation of the inflection from the content word makes the task easier. But for the moment that is just speculation.

Among the relatively simple sentences, the ones that stand out the most are those involving adjectives. Although it is quite good at comprehending them, the model hates to produce adjectives. Part of the reason for this is that the adjective modification rate for subjects was accidentally set to 1/10 of its correct value in Penglish (1% rather than 10%). But even the TRNS-OADJ sentences, in which an adjective modifies the direct object, are considerably more difficult than the TRNS-ONPP, in which a PP modifies the direct object. The reason may be that the majority of NP modifications in Penglish, including PPs, RCs, and adverbs, are post-modifiers. Adjectives are the only pre-modifiers and the model has simply not become used to them. When the model is producing an NP, and has just produced the article, if there is one, the overwhelming pressure is to go ahead and produce the head noun. If adjective modification is needed, this tendency must be suppressed so that the adjective can be produced. The CSCP model has not yet learned to do this, especially when the adjective modifies the main subject. This is clear in looking at the model's word-by-word productions. Given the sentence, "*The old mother thinks,*" the model will tend to begin by producing "*The mother*," leaving out *old*. In word-by-word production, the correct next word, *old* is then given to the network and it is able to correctly produce, "*mother thinks.*"

In producing the conjoined sentences, CONJ-TI and CONJ-IT, the model also has trouble producing the conjunction after the first clause. Rather than producing the conjunction, it tends to produce the article, which would be the next word, a preposition, *that*, or the verb from the second clause. For example, if given the sentence, "*The cop sees the house and a ball goes.*", the model might produce, "*The cop sees the house*", but then follow it with *a*, *on*, *that*, or *goes*. There is also a tendency for elements from one clause to intrude on the other clause. This is particularly true of the verb and direct object. This may be partially due to the fact that these sentences contain one transitive clause and one intransitive. Most of the intransitive verbs are optionally transitive, so the model has a tendency to try to fill the gap with an NP from the other clause.

The SUB-R-TI and SUB-R-IT sentences, in which a subordinate clause appears on the right, are much like the coordinate clause sentences. The model mainly has trouble producing the subordinating conjunction and has a similar pattern of errors in doing so. The SUB-L-TI and SUB-L-IT sentences, in which the subordinate clause appears on the left, are a different story. The model never starts the sentence with the correct conjunction. Once it has gotten past that, it can often make it through the rest of the subordinate clause. But then, rather than continuing on to the main clause, it tries to end the sentence. Throughout the main clause, it will occasionally miss a word, sometimes perseverating a word that has already appeared, but it often just predicts that the sentence will end.

As should be expected following the investigation of relative clauses in Chapter 11, the model finds producing right-branching RC sentences much easier than center-embedded ones. The easiest case is the RC-RO, right-branching object-relative, on which the model sometimes makes no mistakes. Subject-relatives are actually more difficult for the model to produce. This may be because, across the language, the NP VP ordering of an object-relative is more common than the VP NP ordering of a subject-relative. This is particularly true after the word *that* because of the high frequency of sentential complements. The model does seem to have a strong tendency to follow the relative pronoun with a noun or article when trying to produce a subject-relative. When producing sentences with center-embedded RCs, the model has a tendency to try to skip the RC and go straight for the main verb. Once the RC begins, it encounters many of the same problems seen in right-branching RCs. However, it then faces the problem of returning to the matrix clause, which it can do only sometimes. Frequently, the model tries to follow a subject-relative with another RC, as if the object of that clause were itself to be modified.

12.2 Free production

So far we have looked only at word-by-word production, which is essentially prediction in a strong semantic context. Free production is a better reflection of real language production. In this case, the model is not provided with the correct sequence of words, but each word the model outputs becomes the next word in its input. Free production is, in general, a much harder problem. Once the model makes a mistake there is vast potential for things to go wrong. If

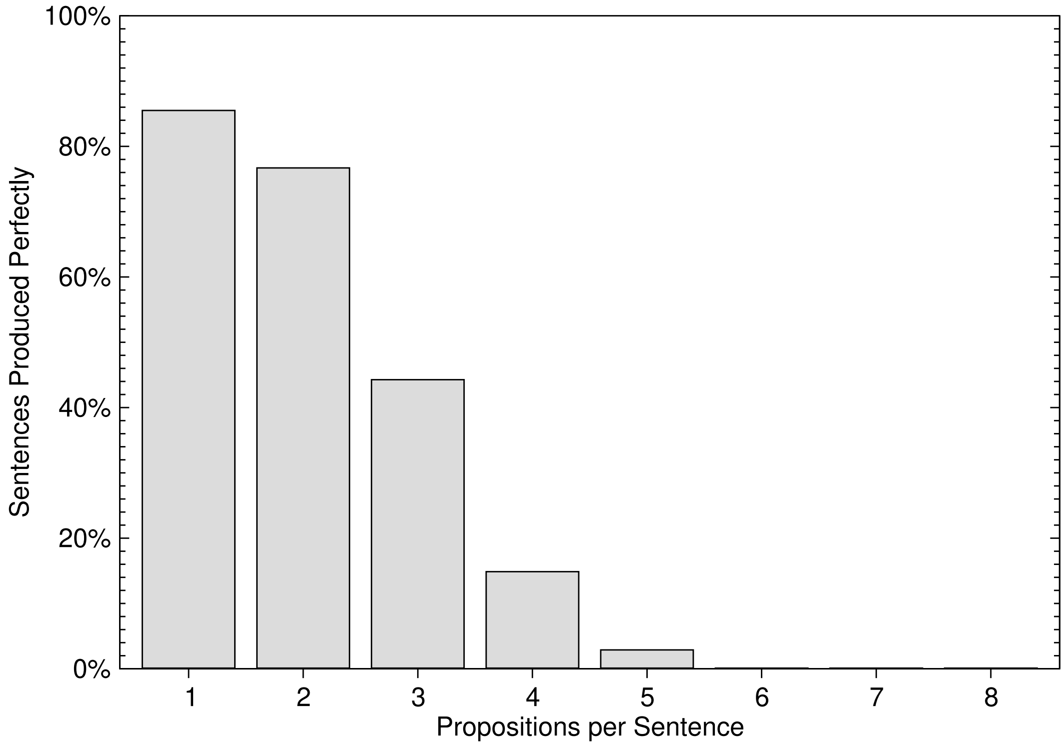


Figure 12.8: Percentage of sentences produced perfectly, with no errors, as a function of the number of propositions per sentence.

the model does not pay enough attention to its prior productions, it could perseverate on a particular word or phrase indefinitely. Earlier versions of the model have been known to do just that. If two clauses share a particular word, or if a word migrates from one clause to another, there is the possibility that the model will fixate on this word and begin producing whatever should follow, even if it is from a different clause.

In other ways, free production may be easier than word-by-word production because the model has the liberty to produce what it wishes and what is coherent to it. If the model has trouble with a certain type of word or phrase, it might drop it and go on with the rest of the sentence. Or, if the model is confused about the proper order of two phrases, it can produce them in the order it likes and then continue on with the rest of the sentence, rather than being forced into the correct ordering, which may add to the confusion.

Free production is more difficult to measure than word-by-word production because the model's productions do not necessarily align with words of the intended sentence. The simplest measurement is whether the model produces the sentence perfectly correct in its entirety. Figure 12.8 shows the percentage of sentences, by number of propositions, which the model is able to produce with no errors. It gets 85.5% of the 1-prop sentences and 76.7% of the 2-prop sentences correct. But then performance starts to decline rapidly, to 44.3% for 3-prop and 14.9% for 4-prop. Although it is very rare, the networks are able to produce a few 6-prop sentences correctly, including such sentences as:

(63a) A teacher told you that the cop gave managers something.

(63b) The boys told you put food on a table quickly.

(63c) The mother thought the boy used a car in a small house.

(63d) The girl told me that the cop was using old cars.

(63e) Something saw that the father is using a old ball for the players.

(63f) I believe the girl knew something by a big lawyer.

Figure 12.9 shows the percentage of perfect sentences across the range of 27 sentence types. As expected from the analysis of word-by-word errors, the model is at a loss when it comes to coordinate or subordinate clause sentences

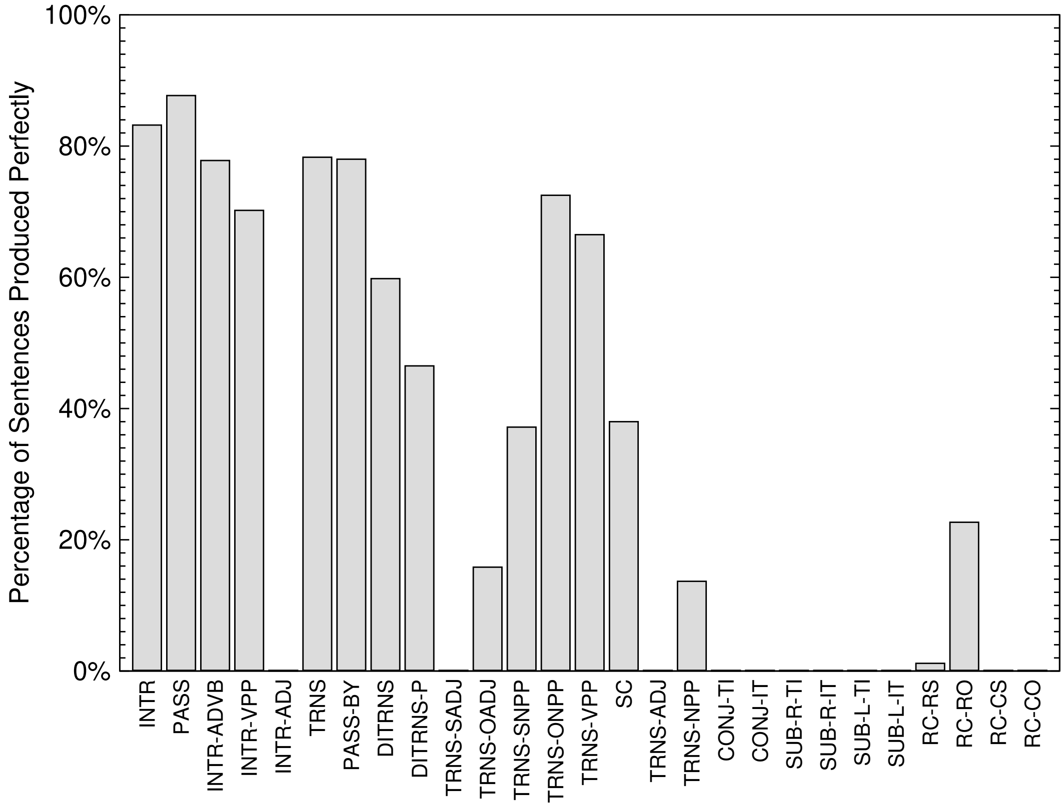


Figure 12.9: Percentage of sentences produced perfectly, with no errors, for a variety of sentence types.

or center-embedded object-relatives. Any subject-modifying adjectives are also a killer. But otherwise, the model has a reasonable rate of success in producing simple transitives, passives, prepositional phrases, adverbs, and sentential complements.

Simply looking at the percentage of perfect sentences does not tell us much about the types of errors committed by the model, nor its ability to recover from those errors. Actually coding the errors by hand would clearly lead to the most useful information, but it is very time consuming and can only be done on a limited basis. Writing a program to do this would be extremely difficult. Detecting word drops or substitutions is fairly easy, but detecting clause reordering or borrowing is much harder. The types of errors detectable by the system would probably have to be hard-coded, which would necessitate determining in advance all possible error types.

In lieu of this, a relatively simple automated method of analysis is to look at the minimal edit distance between the original sentence and the model's actual production. The edit distance algorithm is quite easy to implement and runs quickly as long as a dynamic programming approach is used. In computing the edit distance, word drops, insertions, and replacements were all counted as single operations. Thus, if the model were to exchange two words, the edit distance would be 2, which could either be considered two replacements or a drop and an insertion. It might be reasonable to consider an exchange an atomic operation with cost 1, but that was not done here.

When we compute the average edit distances for different types of sentences, we do not necessarily want to compare them directly. A long sentence is likely to result in more errors than a short one, even if the average difficulty is the same. Therefore, we will instead look at the ratio of the edit distance to the number of words in the sentence. In essence, this ratio is the percentage of errors committed by the model out of all possible errors it could have committed. This will be referred to as the *free production error rate*. In theory, the free production error rate could be greater than 100% if the model just kept babbling. But in practice, it is always less than 100% for a trained network.

An interesting comparison is that between the word-by-word error rate and the free production error rate. If the model is unable to recover from its errors, we would expect the free production rate to be much higher than the word-by-word rate. But if it is generally able to recover, there should be little difference between them. Figure 12.10

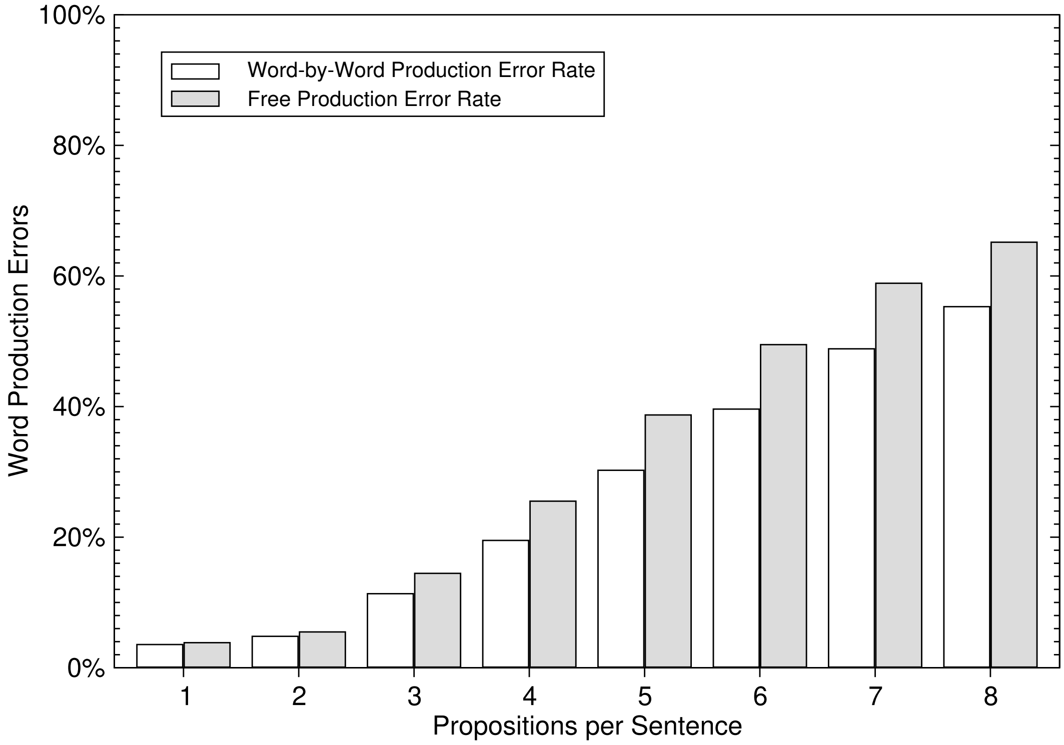


Figure 12.10: Percentage of possible word-by-word and free production errors committed as a function of the propositions per sentence.

shows both the word-by-word and free production error rates as a function of the number of propositions per sentence. Interestingly, the free production error rate is not that much higher than the word-by-word rate. Across the testing corpus, the former averages 33.6% while the latter averages 27.4%. Thus, the free production error rate is about 23% higher than the word-by-word error rate. This is a significant difference, but not a drastic one. The model is clearly not completely thrown off by its mistakes.

Figure 12.11 shows a similar comparison between word-by-word and free production error rates across the sentence types. In general, the free production error rate is just slightly higher than the word-by-word rate. However, notable exceptions to this are the sentences involving adjective modification of the main subject: INTR-ADJ, TRNS-SADJ, and TRNS-ADJ. The reason for this seems to be that the model knows that the subject must be modified, but is unwilling to modify it with an adjective. As a result, it tends to post-modify the subject by fabricating a prepositional phrase. Usually, the object of the preposition is copied from the subject itself or from the matrix object. The following are some typical examples of intended TRNS-SADJ sentences paired with the model's actual free productions. Again, one should keep in mind that this particular impairment is largely due to the unnaturally low rate of adjective modification of subjects in Penglish and the model is much better at adjective modification of objects.

(64a) The new player will take a boy.
The player with a boy will take a boy.

(64b) The nice girls show the mother.
The girls with the reporter show the mother.

(64c) The new manager used a book.
The manager with a manager used a book.

(64d) Loud lawyers gave the picture.
Lawyer gave the picture.

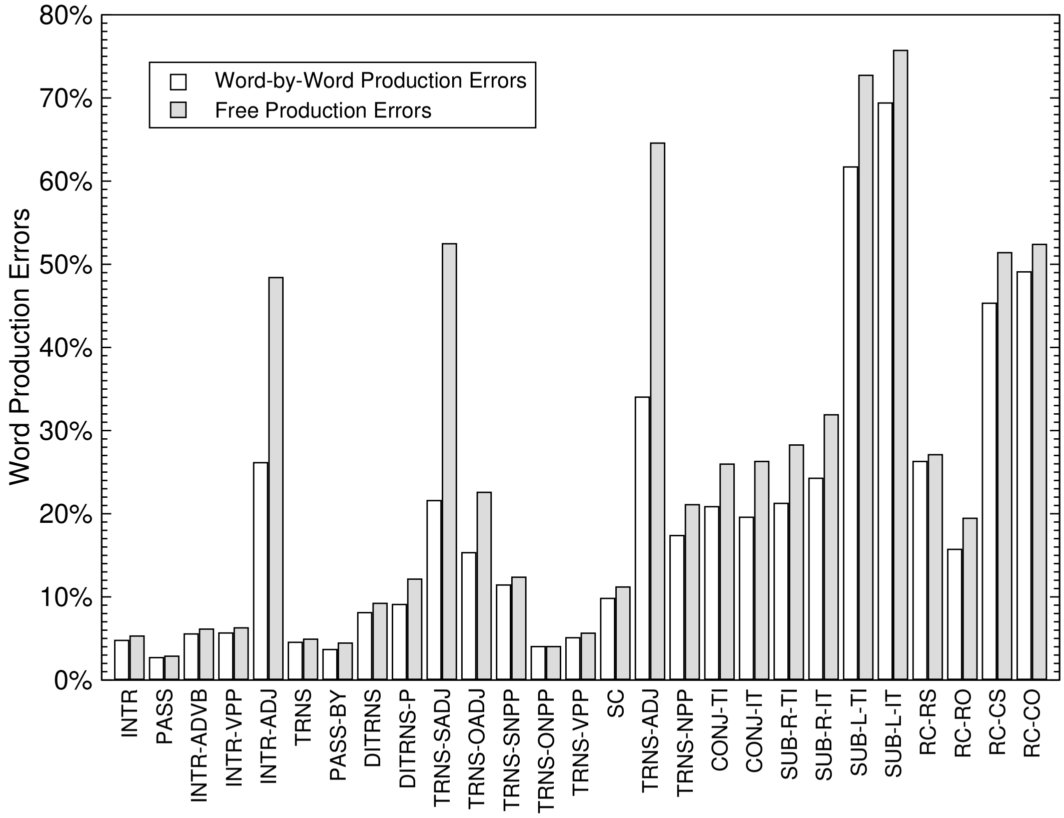


Figure 12.11: Percentage of possible word-by-word and free production errors committed for a variety of sentence types.

12.2.1 Common production errors

Aside from the consistent and profound difficulties the model has in producing certain constructions, such as adjectives, center-embedded relative clauses, and initial subordinate clauses, which have already been identified, it is interesting to consider the types of occasional errors made by the model in producing sentence types that it is generally quite good at. These more subtle sorts of production errors are best analyzed by hand. Because the distribution of errors made by the model is likely to depend on the sentence type, five forms were selected for this analysis: TRNS, DITRNS, SC, TRNS-NPP, and RC-RO. For each sentence type, 100 examples were selected at random from among the free production trials on which the network made at least one error. The errors made were then codified and counted by hand.

Errors were encoded both by their target location and type of error. The possible target locations, or elements of the sentence on which the error occurs, differ among the sentence types. For a simple transitive, the targets of interest would be the subject NP, the object NP, the main verb, and any articles. The articles were not distinguished by which NP they belong to. For other sentence types, targets may also include prepositions, relative pronouns, and any other NPs or VPs. The following are brief descriptions of the error types coded for:

Change: Changing a target to one with a different stem, but of the same class, that does not appear elsewhere in the sentence (*dog* → *cat*).

Borrow: This is the same as a change, but the new stem must appear elsewhere in the sentence. Presumably, the fact that the new word is actually supposed to appear in the sentence increases the likelihood of its replacing the target.

Other: Changing the target to a word of a different class (*dog* → *ate*).

Drop: Leaving out the target word.

Repeat: Repeating the target word or the whole phrase that it heads (*dog* → *dog dog*, *in the house* → *in the house in the house*). If there is more than one, each repetition counts as another error.

Insert: Adding an extra word of the given type.

Number: Producing the wrong number (singular vs. plural) for a noun or the wrong agreement inflection for a verb (*eats* → *ate*).

Tense: Producing a verb in the wrong tense (*is eating* → *was eating*).

Aspect: Converting an active verb to a passive or vice versa.

Case: For pronouns, using an accusative form when a nominative is called for, or vice versa (*us* → *we*).

By working through the frequencies with which these errors occur and seeing some examples of common or interesting errors, the reader should gain a better understanding of the model's production abilities, and well as some ideas about possible mechanisms or shortcuts used by the model in performing free production.

Simple Transitives

The errors made by the model on the simple transitive (TRNS) sentences are generally quite straightforward. Of the 100 sentences on which the model produced at least one error, 84 had only one error, 15 had two errors, and one had four errors. Table 12.1 shows the counts of production errors of each possible type that occurred on these sentences. Empty cells are error types that could not possibly occur, such as a tense error for a noun target. Cells with dashes (–) are possible errors that never actually occurred.

The most common type of error in producing simple transitive sentences is by far the change. Changes are more common on the object than on the subject, but occur most often on the main verb. Changes usually involve semantically similar words that retain number and article. They are, essentially, semantic errors. The target for all of the changes

Error	Subject	Object	Verb	Articles
Change	8	29	41	13
Borrow	1	–		–
Other	–	1	2	–
Drop	–	–	–	1
Repeat	–	–	–	1
Insert	–	–	–	1
Number	1	1	–	2
Tense			5	
Aspect			2	
Case	–	1		

Table 12.1: Free-production errors occurring on 100 TRNS sentences for which there was at least one error.

to articles was either *that* or *these*, which are two of the least common articles, and all of these errors occurred on the subject. The model usually changed these articles to *this* or *a*.

Borrows were very rare on these sentences, leading one to believe that the one case that did occur may very well have just been chance. But as we will see on the other sentence types, borrows are quite common when the sentence involves more objects, but rarely occur between subject and object. Changes to words of a different form class, which would generally be ungrammatical, are also very rare. In two cases, the model produced a preposition in place of the verb and in one case it repeated the verb instead of producing the object. It is not clear if that should be coded as a repeated verb and a dropped object or a changed object.

Drops, repeats, and inserts are also very rare and only occurred one time each on an article in these sentences. Drops of articles are actually a special case, because having no article is a valid option for plurals or mass nouns. Therefore, if an article was left out in a situation in which doing so is grammatical, it was considered a change rather than a drop. Most of these occurred on the mass noun *food*.

Changes in number occurred relatively infrequently, especially considering the ease with which the model could potentially add or remove a -z from the end of a noun or verb to change its number. One change of an article from *these* to *this* was coded as a number error. Tense changes in verbs were more common. Surprisingly, there were two cases of verb aspect changes, from active to passive. As shown in (65a) and (65b), both cases involved changing the direction of the action, rather than swapping the subject and object. In (65b), which is the only TRNS case on which the model made more than two errors, it seems to have converted the verb to the passive and then tried to change the direct object to the object of the verb-modifying PP *of questions*, which would be grammatical in this case, if the number of the object had not been changed as well. Finally, there was an example by one of the networks of a pronoun case error, (65c).

(65a) Players tell something.
Players are told something.

(65b) I wrote questions.
I was written of question.

(65c) I left us.
I left we.

Ditransitives

The double-object dative is an interesting case to study because the model is faced with two objects, the order of which it must get correct. To complicate matters, the change of a single bit in the semantic representation should cause the model to produce a prepositional dative, which has the order of the nouns reversed. For convenience, the indirect object will be referred to as the IO and the direct object as the DO. Table 12.2 shows the error counts for the 100 incorrectly produced DITRNS sentences that were sampled. Prep. stands for prepositions.

Error	Subject	IO	DO	Verb	Articles	Prep.
Change	2	7	33	3	7	
Borrow	3	10	15		15	
Other	–	–	2	–	–	
Drop	–	3	1	–	1	
Repeat	–	–	–	–	–	
Insert	–	–	–	–	–	7
Number	–	4	5	–	–	
Tense				10		
Aspect				–		
Case	–	–	–			

Table 12.2: Free-production errors occurring on 100 DITRNS sentences for which there was at least one error.

As before, errors on the subject were relatively rare. But errors on the DITRNS sentences, unlike those on the TRNS sentences, are quite rare on the verb. The reason for this is probably that the DITRNS sentences only use a limited set of ditransitive verbs, and they all happen to be fairly high-frequency, and thus less prone to errors. Of the three cases of verb changes, two involved changes to other ditransitives. There were, however, a number of tense errors on the verbs.

Change errors were most likely to occur on the DO, but borrow errors were more balanced between the IO and DO. Most of the cases of borrow errors on the IO and DO were from one to the other. Thus, the model would either produce the DO first and then produce it again, or produce the IO twice. These borrowings are generally accompanied by some confusion about the appropriate articles to use. For example, in (66a), one of the networks has moved the DO to the IO position, but retained the IO article. It then repeats the DO, using its original empty article, but using the singular number of the IO. We can, of course, only speculate as to exactly what types of errors led to the result. Another case of a borrowing is (66b), in which *girl*, which plays the role of subject and IO, takes over the DO as well.

One might expect there to be many examples of IO and DO swapping, but they actually seem to be relatively rare. The model is more prone to borrowing than to exchanging the order of the IO and DO. However, three possible cases of swapping did occur. Two networks made the same error in (66c), exchanging either the complete IO and DO, or just their numbers. In (66d), the IO and DO have been swapped, but a preposition has also been added, turning this into a transitive sentence with a PP modifying *answer*. There were six other cases of prepositions being inserted into sentences, four of which involved conversions of the double object dative into a prepositional dative, as in (66e). The other two seemed to be of that sort but had other errors such as dropping of a noun phrase.

NP drops in general were quite rare. Interestingly, the three cases of IO drops were all pronouns, with two drops of *us* and one of *me*. In the two "Other" errors on the DO, the network replaced it with a verb, which was a repeat of the main verb in one case.

(66a) We will give a reporter dogs.
We will give a dogs dog.

(66b) The girl has given the girls the dog.
The girl has given the girls the girl.

(66c) Owners show the mothers the mother.
Owners show the mother the mothers.

(66d) You have read owners a answer.
You have read a answer by owners.

(66e) The owner will get you the answer.
The owner will get the answer for you.

Error	S1	S2	Object	V1	V2	Articles	Compl.
Change	–	2	12	34	1	8	6
Borrow	1	7	1	10	–	12	
Other	–	–	2	1	–	–	–
Drop	–	–	1	3	–	11	5
Repeat	–	–	–	2	–	–	–
Insert	–	–	–	–	–	–	–
Number	1	4	2	–	2	–	
Tense				4	4		
Aspect				–	1		
Case	–	–	–				

Table 12.3: Free-production errors occurring on 100 SC sentences for which there was at least one error.

Sentential Complements

Because they contain two clauses, sentences with sentential complements offer the possibility of verb borrowing errors, as well as confusions between the two subjects. Errors on the SC sentences are shown in Table 12.3. S1 refers to the matrix subject and S2 refers to the SC subject, while Object is the SC object. V1 is the matrix verb and V2 is the verb within the SC. Compl. is short for complementizer, meaning, in this case *that* or *if*, which are optional for most SCs. There were six cases of changed complementizers, all from *if* to the more common *that*, and five cases of dropped complementizers, all involving *if*.

Errors on the subjects were again quite rare, although errors on the SC subject, S2, were more common than those on the matrix subject, S1. The one case of borrowing on S1 was from the S2. Two cases of borrowing errors on S2 were from S1, four were from the object, and one was unclear because S1 and S2 were the same. There was only one borrowing error on the object. It seems that when the model does make borrowing errors, they are mainly from one object to another, and less often between subjects or between a subject and an object. Because there is only one object in an SC sentence, there is no opportunity for object/object borrowing errors. Two of the number errors on S2 were changes from *we* to *I*. The two "Other" errors on the object were repetitions of the SC verb.

The error profiles on the verbs are quite interesting. There were several errors on the main verb, but few on the SC verb. Quite a few cases of borrowing from the SC verb to the main verb were found. Seven of the ten involved verbs that could take a sentential complement, but three did not. Examples of verb borrowing are shown in (67a) and (67b). The fact that the moved verbs changed to the correct tense, and that this change involved a shift in verb stem for these irregular verbs, suggests that borrowing errors are not a surface level phenomenon. That is, borrowing errors do not all result from competition among the prediction output units. If that were the case, the model would have changed *say* to *found* and *felt* to *sees* or possibly *see*. The fact that a change in verb stem occurred suggests that the error was at a deeper representational level.

(67a) You say that the girl found a baseball.
You find that the girl found a baseball.

(67b) You felt the father sees the boy.
You saw the father sees the boy.

Dual noun-modifying prepositional phrases

The TRNS-NPP class of sentences consists of simple transitives with two PPs, one modifying the subject and the other modifying the object. This creates a lot of potential for confusion. To produce the sentence correctly, the model must avoid confusing the four NPs or the two prepositional phrases. It must also be able to handle the fact that the first PP is embedded between the subject and verb. Table 12.4 gives the error counts on these sentences. PPN1 and PPN2 are the objects of the prepositions modifying the matrix subject and object, respectively. "P1" and "P2" are the prepositions themselves.

Error	Subject	PPN1	Object	PPN2	Verb	Articles	P1	P2
Change	2	12	8	7	10	1	14	4
Borrow	–	39	12	8		15	15	4
Other	–	–	1	1	5	–	–	–
Drop	–	1	2	1	3	4	4	2
Repeat	–	–	2	1	1	–	4	6
Insert	–	–	1	–	–	–	–	1
Number	1	7	9	–	4	–		
Tense					1			
Aspect					–			
Case	–	–	–	1				

Table 12.4: Free-production errors occurring on 100 TRNS-NPP sentences for which there was at least one error.

Again there were very few errors on the subject. There were also relatively few errors on the object of the second preposition, PPN2. PPN1 experienced a lot of borrowing errors and a number of change errors. Most of the borrowing on PPN1 was from the main object. The borrowings on the object were about evenly distributed between PPN1 and PPN2, with one or two from the subject.

Occasionally, the errors on these sentences were difficult to encode. The model had a tendency to fabricate prepositional phrases composed of pieces drawn from elsewhere in the sentence. For example, in (68a), the object is changed and a PP, *for the boy*, is inserted that may be a combination of *for me* and *with this boy*. In (68b), PPN1 is borrowed from the main object, then the verb is dropped and replaced with a preposition. Similar sorts of errors accounted for the five "Other" errors on the verbs. Most cases of noun drops were accompanied by additional errors, like duplicated prepositional phrases.

More errors occurred on the first preposition, P1, than on the second. In particular, there were quite a few changes to P1 and borrowings from P2. The repetitions were often difficult to encode. The preposition itself was rarely simply repeated. Generally, the model repeated a preposition and filled in an object to go with it. In (68c) the presence of two *of*'s seems to have overpowered the verb as well. In (68d), there are a number of things going on. The first PP is replaced by something that seems to be a combination of the second PP and the object. Then the object is replaced by *me*. This may be something of a borrowing from PPN1, as *me* and *you* are semantically and syntactically related. Then the model inserts another PP using the main object, which could be interpreted as verb-modifying, and then gets the second PP wrong but finishes up okay. Example (68e) is a rare case in which the model fixated on a phrase until it eventually gave up.

(68a) Something for me gives the plane with this boy.
Something for me gives something for the boy with this boy.

(68b) Lawyers for the father have had tables on a floor.
Lawyers for the table on the table on a floor.

(68c) The teachers of the girl wrote a question of owners.
The teachers of the girl of a question of owners.

(68d) A girl with you asked the owner of a ball.
A girl of a owner asked me for a owner for a ball.

(68e) The story on the table involves the cop with us.
The story with the cop with the cop with with.

Right-branching object-relatives

The final sentence type we will consider is the right-branching object-relative. This is another case in which there are two verbs to confuse, and the verbs tend to be more semantically similar than those in an SC sentence since they are both used transitively. The errors for these sentences are shown in Table 12.5. S1 stands for the matrix subject and S2

Error	S1	Object	S2	V1	V2	Articles	RP
Change	2	9	3	11	10	5	3
Borrow	–	6	9	9	4	5	
Other	–	–	1	4	1	–	8
Drop	–	4	14	–	7	3	38
Repeat	–	5	–	1	2	1	1
Insert	–	–	–	–	–	1	1
Number	–	2	2	1	2	–	
Tense				3	7		
Aspect				1	4		
Case	–	–	1				

Table 12.5: Free-production errors occurring on 100 RC-CO sentences for which there was at least one error.

for the subject of the RC. Likewise, V1 is the main verb and V2 is the RC verb. RP stands for the relative pronoun which always introduced the RC.

Errors on the nouns were less common in these sentences than in some of the other types. There were some of the usual cases of changes and borrowing on the object and on S2. Although drops were not very common on other sentence types, there were quite a few noun drops in this case. The reason may be that there is competition between the correct sentence form and other forms for which the main object or S2 should be left out. For example, consider (69a). By leaving out *evidence*, the sentence takes on a perfectly coherent SC form. Example (69b) is an interesting case in which the relative pronoun was moved and the phrase order in the RC reversed to create an almost coherent SC sentence using the same words.

There were more cases of dropping S2 than of dropping the object. One might expect this to occur if the model were to confuse the object-relative with a subject-relative, and thus begin it with the verb. However, there do not seem to be any errors of this type. The model actually tends to drop the relative pronoun along with S2. The result, (69c), is again something like an SC, although it is often incomplete and the model ends the sentence after the verb as it normally would. Example (69d) shows a pronoun case error, which the model makes only rarely and usually on objects.

There were a number of errors on the verbs, including several borrowings in both directions. There were more borrowings on the first, matrix, verb, but also several drops of the RC verb. There was one case, (69e), in which the two verbs were swapped. The most common error was to drop the relative pronoun. Of course, the relative pronoun is optional with object-relatives, so this is an understandable mistake.

(69a) I saw evidence that the manager was giving.
I saw that the manager was giving.

(69b) You show the ball that the father will have.
You show that the ball will have the father.

(69c) The boys showed cars which schools have had.
The boys showed cars have had.

(69d) The cop has had something that I give.
The cop has had something that me give.

(69e) The girl left the trumpet that I played.
The girl played the trumpet that I left.

Summary

This study of free production errors gives us a somewhat better understanding of the properties of the CSCP model's production mechanism. Errors on the main subject are quite rare. This may be because nearly all sentences begin with a subject NP and the subject thematic role is usually the first proposition loaded into the encoder network. Therefore, the mapping is relatively easy and well practiced. Nearly all change and borrow errors involve words of the same class. Nouns are changed to other nouns and verbs to other verbs. Only on rare occasions is this violated. Drops and insertions of required elements are not very frequent.

When borrows do occur, they tend to involve changes in the stem of the noun or verb, but they often retain the number and tense. Borrowings and changes are also more common when the two words share semantic properties. This suggests that the errors have their root at a fairly high level in the process—at a semantic or lexical level rather than at the surface-form level. Verb borrowing is more common when the two verbs share argument structures. Thus, production errors on verbs generally do not result in syntactic errors.

Borrowing between nouns is generally from one object to another, and rarely involves the sentence subject. This could be because the objects are more similar in terms of thematic role or simply because their representations are noisier than those of the matrix subject. Borrowing is more common from a later word to an earlier one. This was particularly true among the verbs in the SC and RC-CO sentences and among the prepositions in the TRNS-NPP sentences. Thus, repetition of a word that was already produced in its proper place is less common than producing a word too early and then producing it again in the correct position. Furthermore, immediate repetitions of an individual word were also fairly rare and generally occurred only in conjunction with a number of other errors.

These patterns of performance may implicate some mechanisms used by the model to produce sentences. Two requirements of sentence production are keeping track of the state, or where you are in the sentence, and producing the appropriate output when in a particular state. The model seems to be fairly good at the former. If it did have serious trouble keeping track of the state, it would have a frequent tendency either to end sentences early or to repeat itself. If the model thought it had already produced something that it had not, it would end up dropping that part of the sentence. If the model were to forget that it had produced something, it would probably try to produce it again, possibly ad infinitum. Earlier versions of the model, which were not trained as well, were prone to these sorts of errors. But drops and repetitions, at least of the sort one would expect from a "state" error, are quite rare in the current model.

Assuming the model can keep track of the state, the harder problem is producing the appropriate constituents given the state. This is where the model seems to have most of its difficulty. In producing the right constituent at the right time, the model will be aided if it can keep track of the set of constituents that it has yet to produce. In terms of activations, one might think of these as potentiated constituents. Once the model produces a constituent, that item must be dropped from the set of pending items. In other words, it must be suppressed. Given such a short list of possible constituents, choosing the right one for the appropriate state is an easier process. But one would expect that errors will sometimes be made in selecting the proper item from the list. This will tend to result in borrowing errors in which a word is used that should appear later in the sentence. As we have mentioned, the model is indeed prone to these sorts of errors.

Keeping track of constituents in this way can lead to other problems. If the model drops an item from the set too early, that item may never be produced. This situation can explain the large number of change errors on the direct object in the DITRNS sentences. If the model produces the IO, but drops both the IO and the DO from the set of things still to be produced, it will be unable to produce the DO properly and may have to make something up when its slot appears. Another type of error that might be committed is failure to suppress an item, or drop it from the list, once it has been produced. This would lead to borrowings from an earlier word to a later one. However, the model seems less prone to such borrowings than to those from a later word to an earlier one. This suggests that the model is quite good at suppressing the sentence elements it has produced, and its errors may reflect a tendency to over-suppress, rather than to under-suppress. These analyses are, however, highly speculative and, to say anything more definitive about the production mechanism of the model, one would have to conduct a careful study of its internal states. While this sort of analysis is necessary if the model is to be fully understood, it would be quite difficult to do.

12.3 Agreement attraction

One type of production error that English speakers and writers are prone to make on a regular basis is that of incorrect verb agreement caused by a noun phrase interceding between a subject and its verb. If the second, or local, NP differs from the subject in number, particularly when it is plural, there is a tendency for it to co-opt the verb. An example of this, borrowed from Bock and Miller (1991), is given in (70). While agreement-attraction errors can be found in everyday speech and writing, Kathryn Bock (Bock & Miller, 1991; Bock & Eberhard, 1993; Bock, 1995) has conducted several experiments specifically targeted at eliciting these errors.

(70) The time for fun and games are over.

12.3.1 Empirical results

Bock and Miller (1991) performed three experiments on agreement-attraction. In each, subjects listened to a sentence preamble consisting of a subject followed by a PP or subject-relative. They were asked to repeat the preamble and complete the sentence. The subject and the local NP were each manipulated to be either singular or plural, resulting in four conditions which we will refer to as SS, SP, PS, and PP. In the SP condition, for example, the subject is singular but the local noun is plural. In the first experiment, another factor, whether the post-modifier is short or long, was added. Lengthening of the modifiers was generally achieved through the addition of two or three adjectives.

Analysis of the subjects' productions revealed a substantial number of agreement errors, about 4.9% overall. Nearly all of the errors occurred in the SP condition. The 50 errors in that condition accounted for 15.6% of its total responses. There were 7 errors in the PS condition, 4 in PP, and 2 in SS. There were no significant effects of post-modifier length on the error rate.

A second experiment replaced the length factor with a manipulation of the animacy of the local noun and also of the concreteness of the subjects. There were again more errors (29) in the SP condition than in the PS (8), SS (10), or PP (1). The 10 errors in the SS all occurred in the animate local noun conditions. This finding is difficult to explain, but it did replicate in a slightly modified version of the experiment, as did the approximate number of errors in the other conditions. The other effects of animacy and concreteness were rather complex, but essentially there was not much of significance to speak of.

A third experiment looked at a slightly different form of agreement-attraction error. In all of the preambles, the subject was modified by an object-relative clause that was missing its verb. The first words to follow the preamble should form the VP of the RC. Therefore, the correct agreement is with the local noun, not with the matrix subject, although attraction errors are possible. Following this, the main verb should be produced, agreeing with the matrix subject. The animacy of the subject and local noun were also manipulated. In the easy, matching conditions, there was just one embedded verb agreement error on PP and none on SS. But there were 7 errors on SP, and 29 on PS. Thus, in this case there were again more incorrect productions of plural verbs than of singular verbs. Most of the errors occurred with the animate subjects, suggesting that animacy does play a role in this type of sentence. The researchers also gathered production errors on the second, or matrix, verb. Here the error rates were 18 on SS, 56 on SP, 36 on PS, and 3 on PP.

Bock and Eberhard (1993) conducted four experiments on agreement-attraction. In the first two, along with the standard SS and SP conditions, pseudoplural local nouns, such as *cruise* (*crews*), were also tested to see if there is a role played by the phonological form of the local noun, or just its actual plurality. The type of pseudoplurals differed between the two experiments, and the first used auditory presentation of the preamble while the second used visual presentation. The result was no errors in the pseudoplural or SS conditions in either experiment, but a total of 52 errors in the SP conditions.

As a further test of possible phonological effects, the third experiment compared local nouns with regular (*rats*) and irregular (*mice*) plurals. The number of errors elicited was quite small, so a comparison may not be reasonable, but the authors argued that there was no difference between the irregular and regular conditions. This is again evidence that the surface form of the local noun is not what matters. Summing across regularity, the error rates were 1 on SS, 17 on SP, 8 on PS, and 0 on PP. In their fourth experiment, Bock and Eberhard (1993) compared collective local nouns, like *fleet*, with individual ones, like *ship*. All subjects were singular. There was no effect of this manipulation either, suggesting that it is "subcategorical" plurality, rather than "notional" plurality that matters. Summing across local noun individuality, there were 3 errors in the SS condition and 155 in the SP condition.

The results of the experiments reviewed here are shown in Table 12.6. Only the four main conditions are listed, as most of the other variables tested had no effect and issues of animacy will not be addressed here. All values have been converted to percentages of the total responses. Because many responses were either ambiguous or did not immediately continue the sentence using a verb, the actual proportion of errors to verb productions with correct number agreement would generally be a bit under twice as high. The first seven studies listed in the table used preambles in which a PP or subject-relative clause followed the subject. The last two lines are from Bock and Miller (1991), Expt. 3, in which

Study	SS	SP	PS	PP
BM91-1	0.6%	15.6%	2.2%	1.3%
BM91-2a	2.0%	5.7%	1.6%	0.2%
BM91-2b	2.5%	4.9%	1.6%	0.6%
BE93-1	0%	10.3%	–	–
BE93-2	0%	21.9%	–	–
BE93-3	0.5%	8.9%	4.2%	0%
BE93-4	0.4%	20.2%	–	–
Average	**1.1%**	**12.1%**	**2.0%**	**0.5%**
BM91-4 Emb. Verb	0%	1.4%	5.7%	0.2%
BM91-4 Main Verb	3.5%	10.9%	7.0%	0.6%

Table 12.6: Error rates from several agreement-attraction experiments from Bock & Miller (1991) and Bock & Eberhard (1993).

the preamble contained an object-relative clause that was missing its verb.

The results of these experiments seem fairly clear and consistent. Attraction errors are most common in the SP condition. Errors are less common in the PS and SS conditions, and least common in the PP conditions. Thus, it seems that subjects are indeed mislead by a mismatch in number between the subject and local noun, but there is a much stronger tendency to use a plural verb form incorrectly than to use a singular form incorrectly. On first look, this is rather perplexing. Because, as we saw in Section 4.5, singular subjects are just under twice as common as plural subjects, a naive frequency-driven explanation of production errors would predict conversions from plural forms to the more common singular forms. Thus, the human data seems to go against direct frequency-based expectations. It would, therefore, be surprising if the model, which is probably very much influenced by frequency, were able to replicate these effects.

12.3.2 Experiment 12

Basic agreement attraction was tested in the model using transitive sentences with a subject-modifying PP. Five hundred sentences were extracted from a large corpus of Penglish fitting each of the four conditions: SS, SP, PS, and PP. The SP condition, for example, had a singular subject but a plural local noun as the object of the preposition. The main verbs used in the sentences were all unambiguously either singular or plural. Thus, they were generally either in the simple present tense (*plays/play*) or in a present or past progressive tense (*was playing/were playing*). Simple past tense and other ambiguous forms were eliminated.

As with the human subjects, we are interested in what the model produces following the preamble consisting of the subject and PP. Specifically, we are most concerned with cases in which a verb is produced that is unambiguously singular or plural. There are three ways in which this experiment could be conducted using the model. Which of these is the most appropriate method depends on how one thinks the subjects are accomplishing the task. In *free production mode*, the meanings of the sentences could be given to the model and its free-productions collected and analyzed. This is the most appropriate method if one thinks the subjects in the experiments of Bock and Miller (1991) and Bock and Eberhard (1993) are internally constructing a complete sentence meaning and then producing their responses based directly on that meaning. But, using this method, the model is free to make changes to the preamble. An alternative is *word-by-word production mode*. In this case, the model is given the complete meaning in advance, but is also fed the correct words and merely predicts the next one. The difference from the free mode is that the model cannot make a mistake on the preamble. The final way to test the model is in *prediction mode*. In this case, the model is not given the meaning of the sentence. Instead, we just feed the words in the preamble, and then we record its next prediction. Because it is not clear which of these three methods is most appropriate, all of them were tried.

With 500 sentences in each condition, three networks, and three testing methods, there were a total of 18,000 trials in this experiment. Needless to say, the productions were not scored by hand. A script was used to do a simple parsing of the model's output, identify the verb, if there is one, and classify it. There were four possible outcomes to any trial: singular, plural, ambiguous, and other. Ambiguous verbs are simple past tense or other forms that are identical in the

Condition	Singular	Plural	Ambig.	Other
Free Production Mode				
SS	1,404	**16**	10	70
SP	1,360	**53**	7	80
PS	**18**	1,383	18	81
PP	**5**	1,391	12	92
Word-by-Word Production Mode				
SS	1,427	**17**	9	47
SP	1,359	**76**	12	53
PS	**18**	1,416	11	55
PP	**9**	1,427	13	51
Prediction Mode				
SS	1,386	**9**	15	90
SP	586	**86**	435	393
PS	**7**	439	527	527
PP	**0**	1,281	59	160

Table 12.7: Response results with the three testing modes used in Experiment 12.

singular and plural. The "other" category is anything that is not a verb. For example, the model sometimes produced another preposition after the local noun, particularly in the prediction condition.

Table 12.7 shows the results of the three versions of this experiment. Each row contains the 1,500 responses in the given condition. The first thing to note is that the error rate is quite low in all of the conditions, but particularly in the production conditions in which the model was given an actual message to produce. In those two conditions, the model produced the correct verb number in 92.5% of the trials.

The interesting agreement errors are the plural forms in the SS and SP conditions and the singular forms in the PS and PP conditions, which are listed in bold. For those who prefer graphs, Figure 12.12 shows the averaged results in the empirical experiments, taken from Table 12.6, along with the agreement error percentages averaged over the three presentation methods in this experiment. Note that, as in the empirical studies, the condition with the most agreement errors is consistently SP. PS and SS are a distant second, and the fewest errors are in the PP condition. Chi-square tests confirm that the error rate in the SP condition is significantly higher than that in the next highest condition in all three experiments, and the error rate for singular subjects is higher than that for plural subjects in all test methods. The difference between PS and PP is significant when errors are summed over the test methods. The biggest difference between the empirical results and those of the model are that the model's error rate is actually somewhat lower than that of human subjects, particularly in the SP condition. However, the model does replicate the relative ordering of the conditions, as well as the fact that the SP condition is by far the hardest.

It seems that the method used to test the model, whether it is given a message or whether it must formulate predictions based only on the given preamble, does not have a major effect on the results. One final aspect of the results worth discussing is the high number of ambiguous and other errors in the mixed SP and PS conditions when testing in prediction mode. In prediction mode, the model is not given the meaning of the sentence, only the preamble, so it is free to complete it as it sees fit. One valid way to continue such a sentence is to modify the local noun with another prepositional phrase. Prepositions accounted for most if not all of the "other" responses in these conditions and the auxiliary verbs *had* and *will* accounted for most of the ambiguous responses. These are perfectly valid continuations, but it is interesting that they did not occur as often in the non-mixed conditions. One explanation for this is that, when the model becomes confused by a mismatch in number between a subject and a distractor, this tends to weaken any support for producing a particular verb. As a result, the model is more likely to fall back on a common, generic response, like a preposition or an auxiliary verb.

It would be nice if we were also able to replicate the findings of Bock and Miller (1991) involving center-embedded object-relatives. However, due to the severe problems that the model currently has with comprehending and producing CO sentences, which were discussed in Chapter 11, performing such an experiment now would not be very useful. The high error rate would likely wipe out more subtle effects like agreement attraction. Such an experiment will have to

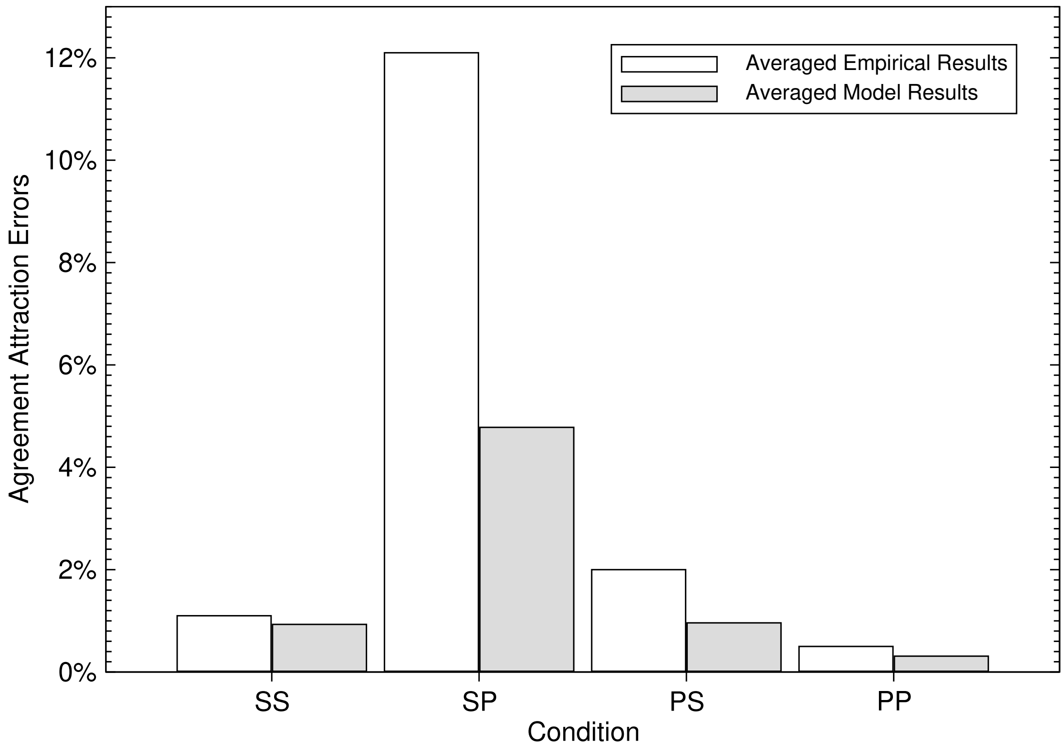


Figure 12.12: Agreement-attraction error rates for human subjects, averaged over the experiments listed in Table 12.6, and for the CSCP model in Experiment 12.

wait until the model's propositional encoding methods for center-embedded relative clauses are improved.

12.3.3 Discussion

That the model is able to replicate the counterintuitive empirical results on agreement-attraction errors comes as something of a surprise. Why should it be more likely to commit errors that result in overproduction of less frequent plural verb forms? The following is one possible explanation. If faced with a task in which one must produce response A to a majority of stimuli and response B only in special circumstances, a natural solution is to make A the default response. The stimulus for A, a singular subject in this case, causes no particular change in the model's state, because it was expecting to produce a singular verb all along. But when the stimulus B occurs, a plural subject, the model must suppress response A in favor of response B. Thus, the model's approach to the problem is asymmetric. The more common stimuli have little impact, because they are expected, but when the prodigal son, a plural subject arrives, there is great fanfare. It should be easy to see how the existence of such a mechanism might explain the pattern of agreement-attraction errors. With a singular subject, a plural local noun has a good chance of impacting the model and causing plural agreement. On the other hand, when there is a plural subject, the singular local noun has little impact and causes fewer attraction errors.

Assuming that this account is the correct one, which, more or less, I believe it is, this tells us something about how we should regard frequency when thinking about complex mechanisms such as the CSCP model. It is not sufficient to say that a model's behavior is governed by frequency. Although a model may be very sensitive to frequencies in its environment, how those frequencies affect its behavior will be dependent on the task and on the particular strategy adopted by the model to solve that task. The strategies developed by a connectionist model to make use of patterns in its environment may, in some circumstances, be quite unusual and unpredictable. That is, at least until we have a better understanding of these models.

12.4 Structural priming

Simply put, structural priming is the phenomenon by which speakers tend to repeat syntactic forms that they have recently heard or used. Although, like agreement-attraction errors, cases of structural priming in natural environments have long been noted, it was only elicited under experimental conditions relatively recently, and is now a topic of some interest.

12.4.1 Empirical results

Bock (1986) asked subjects to listen to and repeat a prime sentence. They were then shown a drawing depicting an event and were asked to give a one sentence description of that event. The question was whether the syntactic form of the priming sentence affected the form of sentence used by subjects to describe the picture. These trials were embedded in filler materials and subjects were unaware of the purpose of the experiment.

In the first study, two syntactic distinctions were examined: actives versus passives and prepositional versus double object datives. Prime sentences used one of the four forms. Bock found that prepositional dative primes resulted in significantly more prepositional dative than double-object utterances and double-object primes resulted in significantly more double-object utterances. Because of the overwhelming bias in favor of active utterances, the priming effects for voice were not as strong. However, there were more active responses to active primes than to passive primes and more passive responses to passive primes. Although all of the elicited prepositional datives used *to*, primes using *for* also resulted in the priming effect, albeit not as strongly as with *to*. This suggests that priming can occur across small variations in surface form.

Two additional experiments varied the animacy of the agent in the active and passive prime sentences and found no effect of prime animacy. This supports the conclusion that priming occurs across variations in meaning. These experiments did find significant priming effects on the production of passives but not on the production of actives, although there were trends in the right direction. They also found significant differences in whether actives or passives were used depending on the animacy of the agent in the pictured event, with human agents resulting in many more active descriptions.

However, Bock et al. (1989) studied active/passive primes with animate and inanimate subjects and did find a significant effect of animacy. In this case, all of the pictured events had inanimate agents. Primes with inanimate subjects, both active and passive, resulted in a higher percentage of active productions than did primes in similar voice with animate subjects. This would appear to contradict the findings of Bock (1986), but the manipulation was not quite the same in this experiment. Bock (1986) varied whether the *agent* of the prime was animate, with an equal split between animate and inanimate patients. Bock et al. (1989), on the other hand, varied whether the *subject* of the primes was animate. The object was always inanimate when the subject was animate and vice-versa. Thus, the lack of an animacy effect on priming in Bock (1986) may have been due to the fact that patients in the prime sentences had mixed animacy. At any rate, it appears that animacy can play some role in structural priming. What is not clear is exactly which conditions should result in priming and which should not.

Bock and Loebell (1990) conducted three interesting structural priming experiments. The first compared double-object primes with prepositional datives and prepositional locatives, both using the preposition *to*. Both types of prepositions effectively primed prepositional dative constructions. This suggests that priming may be insensitive to conceptual differences. However, the difference between a dative and a locative is relatively small and arguably not a major conceptual one.

A second experiment was conducted in which prepositional locatives using *by* were compared with passives having an agent *by* phrase. Control primes were active sentences. Both passives and prepositional locatives primed passive descriptions more than did actives. There was no significant difference between the effects of passives and locatives. This helps confirm that conceptual differences do not eliminate structural priming. However, it is not clear from these two experiments whether priming was based on the surface forms or deep structures of the primes. In both cases, the effective primes shared the same surface form.

In their third experiment, Bock and Loebell (1990), attempted to rule out the importance of surface forms by using two priming conditions that shared surface form but differed in deep structure. Priming conditions consisted of prepositional datives, infinitives, and double-object datives. The prepositional datives and infinitives both used a *to* phrase, but they differed in deep structure. In this case, only the prepositional datives effectively primed other prepositional datives. This was taken to suggest that the important factor in the earlier experiments was similar deep structures, rather than similar surface forms. However, one could argue that the prepositional datives and infinitives did not really share common surface structures. In the one case, the *to* was followed by a noun phrase and in the other it was followed by a verb phrase.

All of the syntactic priming experiments discussed thus far had subjects both comprehend and reproduce the prime sentences. The resulting priming could arise from just one or both of these processes. Branigan et al. (2000) found evidence of priming resulting from comprehension alone. Subjects performed a task in which they described pictures to another person, who was actually a confederate of the experimenter, and selected pictures based on descriptions given by the confederate. The confederate controlled whether he or she used double object or prepositional datives and the form of the confederate's description was found to affect the form of the subject's subsequent description of a different picture. In this case, the magnitude of the priming was much greater when prime and target shared the same verb.

One final issue that has been examined is the persistence of structural priming effects. Bock and Griffin (2000) used similar materials to those of Bock (1986) but varied the distance between prime and target by introducing from 0 to 10 filler items. The priming effect was undiminished even after 10 intermediate items. However, there was a mysterious lack of priming following 4 intermediates. Priming for transitives (active vs. passive) was weaker and shorter-lived than for datives.

12.4.2 Experiment 13

Dell et al. (1999) and Chang et al. (2000) argue that the persistence of structural priming suggests that it may be due to an implicit learning mechanism rather than to activation-based memory. With the support of simulations from a connectionist model, they propose that structural priming operates through the same mechanism by which the model learns to produce sentences. This seems like a reasonable proposal to me, so I will concur and, on the basis of this assumption, conduct some priming experiments on the CSCP model.

Priming trials on the model consisted of training it on a prime sentence, in word-by-word production mode, and then testing its free production on a target sentence. As in Dell et al. (1999), the semantic representation of the target sentence was manipulated to make it ambiguous between the two possible variants. Originally, the intent was to test both double object/prepositional dative and active/passive cases. Creating an ambiguous target for the ditransitives is easy because their semantic representations differ in a single bit, which indicates whether the recipient is expressed with a prepositional dative. This unit can be set to 0.5 to create an ambiguous message.

Unfortunately, due to a poor choice in the way passives were encoded, they differed from the actives in more than a single bit. In both passives and actives, the proposition encoding the subject was given to the message encoder before the proposition encoding the object. But an active and a passive sentence with similar meaning share the same agents, not the same subjects. Therefore, the agent and patient are encoded in different order in a passive than in an active, and it is this order that the model mainly relies on in determining which aspect to use. I wasn't able to find any way to produce a sentence meaning for a target item that is ambiguous between an active and a passive. Averaging the representations at the message layer results in an incoherent message. Therefore, only ditransitives could be investigated in this study. Active/passive priming should be possible with a revised semantic encoding scheme.

While the model was "training" on the prime, a learning rate of 0.2 was used, with no momentum and a weight update immediately after the prime. For simplicity, the model's weights were reset to their pre-primed values after each trial. This prevents cumulative effects and simplifies the experiment as it can be conducted in blocks rather than with balanced, intermingled trials. Presumably the results should not be too different. The target items in this experiment were all simple ditransitives in which the indirect object or the object of the preposition played the recipient thematic role.

The messages used for the prime items were similar to the target items in that the PP bit was set to 0.5 to render them structurally ambiguous. The 200 target sentences were extracted from a corpus of Penglish. Results from the three networks were combined, resulting in 600 trials per condition. The model's productions for the targets were classified as either double objects, prepositional phrases, or errors. Only syntactic violations, or sentences not fitting the double object or PP structure, were considered errors. Lexical errors were ignored.

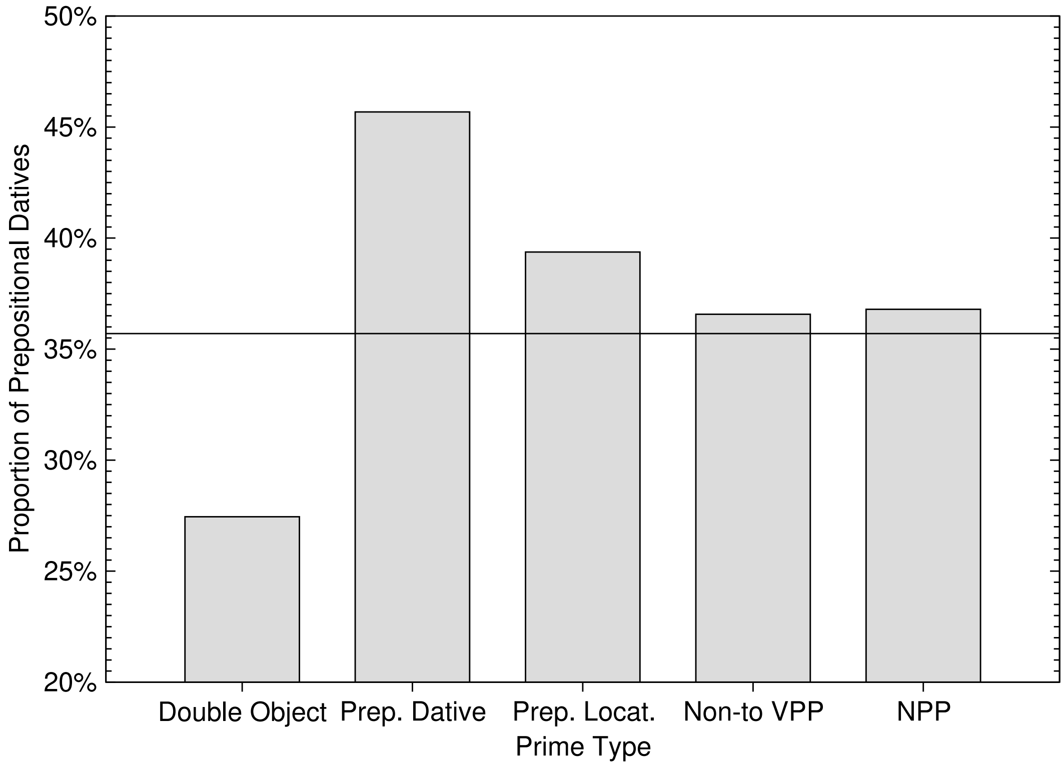


Figure 12.13: Percentage of prepositional datives produced in response to different primes in Experiment 13. The horizontal line is the unprimed base rate.

The model was first tested with no primes to obtain a baseline rate of prepositional dative productions, which was 35.7%. In computing the percentage of PP responses, only the PP and double object responses were considered, leaving out the 40 trials (6.7%) with errors. This practice was used across all of the conditions. The rate of errors was quite stable across the conditions, ranging from 5% to 7%. These differences were never significant and will not be reported.

In the first set of conditions, the model was primed using either double object datives or prepositional datives. The double object datives resulted in a decrease in PP productions to 27.5%. The prepositional dative primes, which all used the preposition *to* and expressed the recipient role, resulted in an increase in PP productions to 45.7%. These effects are shown in the first two bars in Figure 12.13. The base rate of PP productions is indicated by the horizontal line.

When learning is enabled, the CSCP model is clearly capable of experiencing the effects of structural priming. However, these are the strongest conditions, in which the prime type matches one of the two possible renderings of the target. What happens if we use primes that have a verb-modifying locative, expressing destination, rather than recipient, but still using the preposition *to*? As shown in the middle bar of Figure 12.13, the degree of priming is reduced, but is still substantial. This does not exactly match the results of Bock and Loebell (1990), who actually found numerically more priming in the locative case, but I would argue that the model's performance seems more reasonable and may be the more likely outcome if an empirical replication were conducted.

The prime type can be moved farther from a prepositional dative if we use prepositional phrases that express a role other than recipient and do not use the preposition *to*. We can also use noun-modifying prepositional phrases, which are structurally different from the verb-modifying ones and also do not use the preposition *to*. As shown in the rightmost bars in Figure 12.13, these conditions result in only very small degrees of priming. The fact that there is little difference between these two prime types suggests that the model may be mainly sensitive to surface forms in its priming rather than deep structure, but this is a relatively weak test of that.

As mentioned earlier, Bock and Griffin (2000) found that the priming effect was able to last over 10 intervening

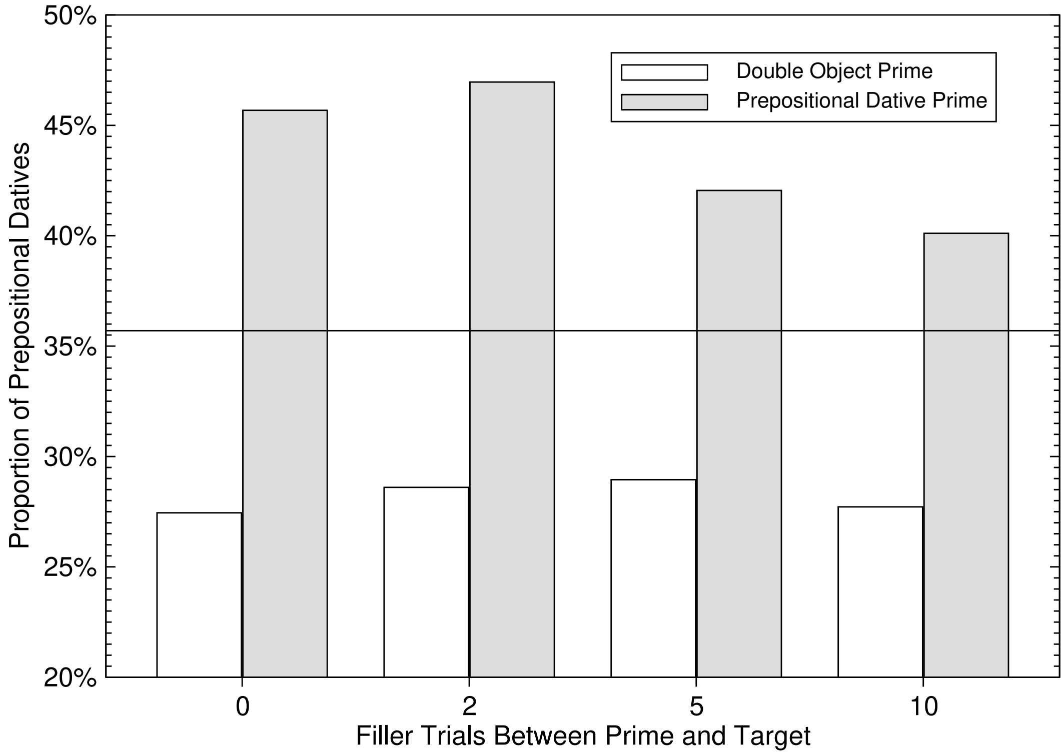


Figure 12.14: Percentage of prepositional datives produced in response to double object or prepositional dative primes with between 0 and 10 filler trials between prime and target. The horizontal line is the unprimed base rate.

filler items. The effect of delays such as this were tested in the model by adding 2, 5, or 10 additional training items between prime and target. These filler items were selected at random from a set of 1,600 sentences consisting of 400 INTR, 400 INTR-ADJ, 400 TRNS and 400 TRNS-SADJ sentences.[1] Importantly, there were no ditransitives or prepositional phrases among the fillers.

Results for the three delay lengths, using both double object and prepositional dative primes, are shown in Figure 12.14. The effectiveness of the double object primes diminishes very little even with 10 filler items. The effectiveness of the prepositional dative primes does seem to diminish somewhat, but priming is still robust even at the longest delays. It is possible that the filler items used here interacted more with the prepositional datives than with the double objects, resulting in more interference for the former.

One final question to consider is where in the model does the learning that goes on during priming have its effect. Does the comprehension side of the model play any role, or do all of the adaptations occur on the production side? To answer this question, the basic priming experiment was repeated with weights frozen on either the comprehension or production half of the comprehension/production system. With all weights updating, the priming effect, or difference in percentage of prepositional datives produced following double object and prepositional dative primes, was 18.2%. When weight updates are eliminated in the comprehension half of the system, this priming effect diminishes slightly to 15.3%. When weight updates are eliminated in the production side of the system, but allowed in comprehension, the priming diminishes to 5%. Therefore, as might be expected, the production side of the system is primarily responsible for the priming effects, but it is not solely responsible, and it is still possible to achieve structural priming with no changes allowed in the production side of the system.

The CSCP model is able to account for several key structural priming phenomena using a mechanism quite similar to that employed by Dell et al. (1999). The model shows robust priming to double object and prepositional datives, which diminishes only slightly over delays of up to 10 filler items. It also demonstrates somewhat weaker, though still significant priming from prepositional locatives to prepositional datives, and small levels of priming from other verb-

[1] See Section 7.3 for descriptions of these sentence types.

modifying and noun-modifying forms. Unfortunately, some other empirical experiments could not be replicated in the current model. Priming based on PPs expressing a recipient but using the preposition *for* could not be tested because *for* only encodes the purpose thematic role in Penglish, which, in this language, bears no similarity to a recipient. Likewise, infinitives, as used by Bock and Loebell (1990), do not exist in Penglish. Because of the way in which their messages are encoded, priming of actives versus passives also could not be tested.

12.5 Summary

Although the CSCP model is quite good at producing simple sentences and those with certain complexities, such as prepositional phrases, indirect objects, and sentential complements, it runs into some serious trouble when producing most complex sentences, and it is even derailed by seemingly simple adjectives. There are several ways in which the model's production performance could be improved. A simple one is to increase the percentage of training trials on which the model is given the correct message in advance. The model was only provided with a complete, full-strength message on 20% of the training trials. Increasing this percentage would weight the model's performance more in favor of production over comprehension. Another possible change is to use a stronger error measure during training. Because of problems with numerical overflow, sum-squared error was used on the *prediction* layer, rather than divergence. Divergence has the advantage that it really penalizes the network strongly when a prediction is very wrong, and generally results in better performance on prediction tasks such as this.

Nevertheless, the model's production behavior has some interesting characteristics which, eventually, we may be able to compare with those of human speakers. Although the word stem and ending are actually produced by separate groups, which collectively make up the prediction layer, the errors on word stem and ending are highly correlated, suggesting that they arise from a single mechanism. When the model does make an error in producing the word stem, it tends to be much less sure of itself than when it is correct. This suggests that its erroneous responses will be sensitive to error signals and likely to improve with further training. Interestingly, the model's free-production error rate is not that much higher than its word-by-word error rate, indicating that it is generally able to recover from its free-production errors.

The model tends to make production errors in the middle of sentences, rather than at the beginning or the end. Detailed classifications of the model's errors on several different sentence types revealed some interesting properties of its behavior. The model seems to make few "state" errors, in which it simply loses track of where it is in the sentence. It also rarely commits errors that involve changing a word to one of a different syntactic class. Most of the errors are either semantic errors, in which a word stem is changed to another one with similar meaning, or borrowings, in which a word is copied from one position in the sentence to a different position. Most of the cases of noun borrowing were between two objects and usually it is the second object than is produced too early, rather than the first object that is repeated. Interestingly, borrowing errors usually retain the number and, in the case of verbs, tense of the correct word for that position. It is just the word stem that is borrowed. This suggests that these errors occur on a fairly deep level. This is not a case of competition between surface forms, but of competition between semantic or lexical forms.

Surprisingly, the model was able to replicate not only the occurrence of agreement-attraction errors, but the fact that such errors are much more common when they involve replacing a singular verb with a less-frequent plural form. This shows that, although the model is sensitive to frequency, the way this will manifest itself is not always so obvious. A plausible explanation for the fact that both humans and the model are more likely to commit plural agreement-attraction errors is that singular nouns and verbs are treated as a default state, which is overridden by an exceptional plural subject. When a plural noun intervenes between a singular subject and its verb, the verb number is sometimes overridden by mistake.

Finally, the CSCP model is capable of exhibiting structural priming that is able to survive over a number of filler items and that can result from primes that differ from the target in underlying or surface form, although priming is much weaker when there is a mismatch in surface forms.

Chapter 13

Discussion

This thesis has introduced the Connectionist Sentence Comprehension and Production, or CSCP, model of human language processing. The model is an extension of earlier connectionist work, with applicability to a broad range of sentence processing phenomena. The goal of this project was certainly not to provide a complete account of all aspects of human language processing, nor to argue that the current model is a sufficient account of all empirical results to which it has been applied. The goal was simply to design and train an initial version of the CSCP model and to evaluate its successes and failures, as well as to begin analyzing the principles on which it operates. Since no model can be perfect, the failures of the model can be at least as interesting and informative as its successes. However, a reasonable degree of descriptive adequacy is necessary for a model to be considered worthy of further investigation. I will leave to the reader the question of whether the CSCP model has achieved the requisite descriptive threshold to motivate it as representing a viable theory of human sentence processing.

This discussion will begin by reviewing the experimental tests of the model's behavior that were presented in Chapters 7 through 12. Section 13.2 summarizes the accomplishments and advancements of the model on a more general level, while Section 13.3 reviews some of its problems and suggests ways in which they could be ameliorated. Section 13.4 is an attempt to clarify the distinction between the implemented model itself and the more general theory that it embodies. Finally, Section 13.5 introduces some intermediate-level principles useful for reasoning about why the CSCP model performs as it does, and it illustrates how the principles arise from the more basic processes with which the model operates.

13.1 Summary of results

This section summarizes the experimental results obtained in the preceding chapters. Chapter 7 included various tests of the model's comprehension performance. The general measures presented in Section 7.1 demonstrated the model's improvement in comprehension performance over the course of training and as a function of the number of semantic propositions composing the sentence meaning. As in other connectionist models, there is rapid improvement in the early stages of learning which tapers off with further training. As should be expected, the less common and more complex sentences with a large number of propositions result in higher error rates, but the model still performs well above chance even on the more complex sentences.

As is apparent in the separation of question-answering errors due to the message encoding and decoding process from those due to the full comprehension process, both sources of errors can contribute to comprehension failures. Errors on relatively simple sentences tend to occur mainly in the mapping from surface form to meaning, but errors on the more complex sentences are as much, if not more, attributable to an inability to fully encode the propositional content of the message and answer questions about it. Theories of human comprehension tend to consider only the possibility of comprehension errors arising in the mapping from surface form through some representation of deep structure. But the current model predicts that errors can also arise in the process of maintaining the propositional content of a sentence in working memory and using it to answer comprehension questions. As will be discussed later, the errors that arise in these two conceptually, if not mechanistically, separable processes are not altogether random, but are sensitive to somewhat different sets of factors.

A challenge that is frequently, and rightfully, put to connectionist models is that they should be able to handle not just the items, or sentences in this case, on which they were trained but should generalize to novel items from the environment. Earlier connectionist models (e.g., Elman, 1991), were trained on languages with small vocabularies and limited syntactic complexity. As a result, the training sets generally included a substantial proportion of all possible sentences in the language. In contrast, the number of natural language sentences encountered in the lifetime of a typical speaker is dwarfed by the total number of sentences possible in the language. Because of its greater syntactic and lexical inventory, relative to earlier artificial training languages, Penglish has many more possible sentences and thus a greater proportion of novel testing sentences. Of the 4-proposition sentences in the randomly generated testing set, only 2% also appeared in the networks' training set. As shown in Experiment 1, the model is as good or nearly as good at comprehending novel 4-proposition sentences as it is at comprehending familiar ones, which we may expect to be slightly easier due to a bias toward higher frequency words and structures. Therefore, the model seems able to generalize, in at least this important way.

Experiment 2 compared the comprehensibility of 25 different sentence types, ranging from simple intransitives to those with center-embedded object relatives. The model's strict criterion question-answering error rate was below 10% and its multiple-choice error rate was below 2% on nearly all of the simple sentences. The easiest sentences for the model are the basic transitives. Surprisingly, the intransitives are significantly harder, but are made easier when the verb is post-modified. This seems to be the result of strict processing limitations placed on the model and will be discussed further in Section 13.3. Passive sentences with a by-phrase expressing the agent are more difficult than actives, which may be a function of their lower frequency, greater length, and non-canonical word order.

The model is quite good at comprehending prepositional phrases and adjectives modifying the direct object, although adjectives modifying the subject cause problems because they were very rare due to a bug in the Penglish grammar. According to the model, prepositional datives are easier to comprehend than double object datives, most likely because many of the latter are temporarily ambiguous. Comparison of ambiguous ditransitives with those using verbs having disjoint sets of direct and indirect objects, which should not lead to ambiguity if the model is sensitive to this semantic distinction, revealed that the latter were significantly easier, despite their lower frequency. Therefore, the model is able to use semantic information to help resolve temporary structural ambiguities.

Sentences involving two prepositional phrases or two adjectives are considerably harder for the model than those with one of each. This suggests that the model is subject to confusion when two structures share similar properties, at either a syntactic or propositional level. As should be expected, it has more difficulty on most complex sentences than it does on the simple sentences, but it is still able to comprehend many complex sentences reasonably well. Sentential complements, in particular, present little problem. Conjoined and subordinate clause sentences are harder, but the multiple-choice error rates are quite low unless a subordinate clause precedes the main clause. These sentences are presumably difficult because of the long-distance attachment that is required between the subordinating conjunction, which starts the sentence, and the verb in the main clause. Some of these sentences also involve the NP/0 ambiguity because the clause boundary is never marked. The model has considerable trouble with center-embedded relative clauses, but this may be attributable to the poor design of their propositional representation, which will be discussed further on.

The majority of the sentence type comparisons just discussed were made without reference to any particular empirical data. However, Caplan and Hildebrandt (1988) did test the comprehension abilities of some aphasic patients on several of these sentence types. Experiment 2b compared the model's performance with that of the patients and, with some minor exceptions, found a substantially similar ordering of difficulty, with a correlation of 0.81. The primary difference, aside from the lower overall error of the model, seems to be the model's disproportionately high error on the center-embedded object relative clauses, which was just alluded to.

Experiment 2c compared intransitive and passive sentences that differed, at the surface level, only by the inflectional morphology of the verb. Nevertheless, in assigning the correct thematic role for the subject, the model had just 12.1% errors for the passives and 1.1% errors for the actives. It is, therefore, able to use information provided by inflectional morphology to correctly interpret otherwise ambiguous structures.

Section 7.4 presented two investigations of the model's ability to correctly resolve lexical ambiguities. Experiment 3a tested words that can serve as either nouns or verbs. The results were somewhat complex due to concerns about lexical frequency confounds. But a reasonable conclusion seems to be that, according to the CIE and CME error measures, which attempt to isolate the errors due to comprehension from those due to propositional encoding, error rates for lexically ambiguous nouns and verbs were in line with those for their unambiguous neighbors. Thus, the fact that some surface forms can inhabit multiple word classes is not a serious hindrance to the model. Experiment 3b tested comprehension of the verbs *nice* and *hard* which have distinct senses that are dependent on the semantics of the modified noun. Note that this differs from the noun/verb ambiguity in that it is not a syntactic one and cannot be resolved until at least the next word is revealed. In this case, as well, the model encountered little noticeable difficulty.

Experiment 4 examined sentential complement sentences ending in a temporal adverb which, due to the verb tenses, was unambiguously either high- or low-attached. In accordance with the pattern of difficulty experienced by human readers, the model showed a strong preference for low attachment. This is partially attributable to frequency. Although the high and low attachment frequencies do not differ all that much in Penglish for the adverb *yesterday*, there is an overwhelming tendency for an adverb, in general, to modify the immediately preceding verb. Another factor that presumably contributes to the difficulty of high attachments is locality, or the lack thereof.

In Experiment 5, sentences were constructed having potentially ambiguous PP attachments to either the main verb or its direct object. These were disambiguated either by the preposition itself or by the noun following it, or they were globally ambiguous. Even in the globally ambiguous case, the model was able to deduce the correct site of attachment in about 74% of the cases, probably as a result of sensitivity to weak statistical biases. When the attachment site was disambiguated only by the semantics of the noun following the preposition, the error rate dropped to 11%. Therefore, the model is able to use semantic information to resolve structural ambiguities. When the preposition itself was the disambiguator, performance was better still with 4% errors for NP attachment and only 0.5% errors for VP attachment.

Sections 7.8 and 7.3.2 presented evidence that the three instances of the model, which differed only in their random initial weights, developed significant variance in their abilities to process particular sentence types. Interestingly, these individual differences arose despite the fact that the networks have identical architecture, training parameters, and experience. This suggests that the random initial state of the networks resulted in unique tradeoffs in the representations or processes that they developed, resulting in potentially substantial ultimate differences in their ability to comprehend certain sentence types. It is possible that model's like this may eventually be able to explain a range of individual differences among normal language users that are not dependent on their innate endowment, nor on experiential differences. But when we are only concerned with average human behavior, it is important to compare this not to that of a single instance of the model, but with the model's behavior averaged over several networks.

Current empirical findings seem to indicate that comparison of the comprehensibility of single relative-clause sentences should result in an advantage for subject-relatives over object-relatives and for center-embedded over right-branching structures (Gibson et al., in press). An earlier version of the CSCP model exhibited just this pattern of behavior. However, as shown in Chapter 11, the current model does not replicate these findings. In fact, it seems to have a significant preference for right-branching structures and for right-branching object-relatives over right-branching subject relatives. The major factor contributing to its preference for right-branching structures seems to be that, when the propositional representations of center-embedded sentences were encoded, the propositions related to the relative clause intervened between those expressing the subject and object of the main clause. Splitting the main clause in this way is unusual for the model and seems to result in serious disruption of its ability to encode the messages of center-embedded sentences. This was not the case in the earlier version of the model, which did show a preference for center-embedding.

Another factor that may be relevant to the processing of RCs is the disproportionately high rate of sentential complements in Penglish. Because the word order, *that NP VP*, in a sentential complement is similar to that in an object-relative, the model may experience some transfer between the two, resulting in faster reading times for object-relatives than would otherwise be expected. Having a verb phrase immediately after the relative pronoun *that*, as in a subject-relative, is actually the more unusual situation. This problem should be reduced by properly balancing the overall frequency of sentential complements and relative clauses in Penglish.

13.1.1 Major structural ambiguities

In addition to the double object datives, optional intransitives, and PP attachment ambiguities just discussed, three major temporary structural ambiguities were studied in Chapters 8–10, including the main verb/reduced relative (MV/RR), noun phrase/sentential complement (NP/S), and subordinate clause (NP/0) ambiguities. Although the findings for each of these were summarized at the end of their respective chapters, a shorter summary is given here.

Three factors were examined in conjunction with the MV/RR ambiguity: whether the relative clause is reduced, whether the verb has an unambiguous past participle form, and whether the subject is a plausible agent of the potentially ambiguous verb. In its comprehension performance, the model was sensitive to all three of these factors in the expected direction, with improved performance when the factors suggest increased likelihood of a relative clause. There was also a significant interaction between verb-form ambiguity and reduction. A meta-analysis showed that the model performs better on obligatorily transitive verbs than on optionally intransitive ones.

Reading times were assessed on the ambiguous verb and at the disambiguation point. If the model is garden-pathed, we should expect fast reading of the verb, which is being interpreted as a main verb, but slow reading at disambiguation. If the model is not garden-pathed, reading times on the verb are likely to be slower because the model must, at that point, begin to establish the parsing structures necessary to deal with an embedded passive relative clause. Analysis of its reading times indeed show that the model is slower on the verb when the RC is marked, when the verb is unambiguous, or when the subject is a bad agent of the verb. There is actually an interaction between reduction and agent plausibility such that the verb is read faster for good agents when reduced but for bad agents when marked. Opposite patterns were seen in its reading times at disambiguation. These were slower for reduced than for marked relatives and for ambiguous than for unambiguous verbs. In accordance with MacDonald (1994), optionally intransitive verbs are read faster on the verb and on the by-phrase but slower on subsequent regions. One problem with the model's performance was the lack of a large garden-path effect at disambiguation for the reduced, ambiguous condition with a good agent. This may be due to several possibilities, including that the model is simply failing to recover from the garden-path in this case. Therefore, we do not see the slowdown that, in humans, presumably reflects reanalysis or reparsing. This issue is returned to in Section 13.3.

The NP/S ambiguity, discussed in Chapter 9, occurs whenever a sentential complement is reduced and the main verb can alternatively take an NP complement. Four factors were studied in conjunction with this ambiguity: reduction, the bias of the verb toward sentential or NP complements, whether the ambiguous NP is a plausible object of the verb, and the length of the ambiguous NP. In terms of comprehension, there were significant main effects of each of these factors in the appropriate direction. There were also strong pairwise interactions between reduction and verb bias, verb bias and NP plausibility, and reduction and NP length, and weak interactions between reduction and NP plausibility, verb bias and NP length, and NP plausibility and NP length.

Reading times were analyzed both on the ambiguous NP and on the disambiguating verb. On the article of the NP there was a reduction effect, but on the noun itself there was a weaker reverse reduction effect. The net result is a moderate reduction effect across the NP. The majority of experimental studies that were reviewed also found reduction effects in the ambiguous region. On the article there was a reverse verb-bias effect (SC-biased is slower than NP-biased), which is stronger in the reduced case. A similar effect was found across the ambiguous region in several empirical studies. Other studies, however, found inconsistent effects of verb bias in the ambiguous region in different conditions.

On the ambiguous noun, there was a strong reverse effect of NP plausibility, and an interaction between verb bias and NP plausibility, such that the slowest condition was the NP-biased with a bad object and the fastest was the NP-biased with a good object. Pickering et al. (2000) also found a reverse plausibility effect at this point. In the disambiguation region there was a large effect of reduction which interacted with verb bias. The slowest condition was reduced and NP-biased, followed by reduced and SC-biased. There was little or no effect of verb bias in the marked conditions. The reduction and verb-bias effects, as well as the greater verb-bias effect in reduced conditions, were supported by most of the empirical studies that were reviewed.

The model experienced a relatively weak effect of NP plausibility at disambiguation. However, it seemed to interact with verb bias such that there was a stronger plausibility effect for the NP-biased verbs. The empirical data generally supports the finding that good objects result in slower reading at disambiguation, and that the effect is stronger, or may only exist, for the reduced NP-biased conditions. The effect of lengthening the NP on reading times at disambiguation depends on whether the SC is reduced. Long NPs result in slower reading in marked conditions but faster reading in reduced conditions. A possible explanation proposed for this somewhat strange outcome is that lengthening the NP results in noise or confusion that leads to a weakening of commitment to the preferred interpretation, which is reflected in a regression to the mean in terms of reading times. This may explain the finding of Holmes et al. (1989) that longer NPs lead to slower reading at disambiguation for SC-biased verbs but faster reading for NP-biased verbs.

The third major ambiguity that was studied here, the NP/0, occurs when an intransitive subordinate clause with an optionally transitive verb precedes the main clause, with no explicit phrase boundary. Two factors were tested in Experiment 8, the first having three conditions: whether the ambiguous verb is intransitive, optionally transitive and biased toward the intransitive reading, or optionally transitive and biased toward the transitive. The other factor is whether the NP following the verb is a good or bad possible object of the transitive form of the verb. In terms of comprehension performance, the strongly transitive verbs were harder than the weakly transitive, despite the higher frequency of the former. Also, there was a significant advantage for the bad objects. Against expectations, the intransitives were actually relatively hard. This may reflect the model's general problems in dealing with intransitive verbs.

The reading-time results matched the consistent empirical findings in the literature. In the ambiguous NP region, intransitives were slower than weak transitives, which were slower than strong transitives, and bad objects were slower than good objects. At the disambiguation point, these effects are reversed, with strong transitives and good objects resulting in larger garden-path effects, and intransitives and bad objects in faster reading times.

Experiment 9 evaluated the model's grammaticality ratings of a series of variations on the NP/0 sentences that were tested by Ferreira and Henderson (1991). Overall, there was a moderately strong correlation between the model's and subjects' ratings, but there was at least one major difference. Although the early closure sentences (starting with an intransitive subordinate clause) resulted in longer reading times at disambiguation, the model did not find the early closure sentences to be less grammatical than the late closure ones (starting with a transitive subordinate clause), as Ferreira and Henderson found. This may have been due to the fact that the late closure sentences used in testing the model had a high potential for semantic confusion because they used similar NPs. Another contributing factor may be that the model was simply not garden-pathed sufficiently by the early closure sentences because it was never provided with explicit clause boundary markings, as are commonly given in English punctuation or prosodic phrasing, and thus did not come to rely on such boundaries to aid disambiguation. The prediction is that the addition of a clause boundary marker as a useful but not entirely reliable cue will result in a greater garden-path effect on early closure sentences when that cue is absent.

Experiment 10 tested the model's susceptibility to incomplete reanalysis when recovering from a temporary garden-path on the early-closure sentences, which was documented in human subjects by Christianson et al. (2001). The model replicated the fact that the frequency of believing that the subject of the main clause is also the object of the subordinate clause verb is much greater for early closure sentences. A second experiment used sentences in which the subordinate clause either preceded or followed the main clause, to control for a potential confound. The model successfully replicated the fact that confusions are less common when the subordinate clause follows the main clause, but still seem to occur at a significant rate in this case. This suggests that the model, like humans, is influenced both by syntactic ambiguity and by potential semantic confusions that are unsupported by syntax. This is another indication that comprehension errors can arise at both syntactic and semantic levels.

13.1.2 Production

Chapter 12 evaluated the model's ability to produce sentences given an intended message. Two forms of production were considered: word-by-word and free production. The model is quite good at producing simple sentences and those with prepositional phrases, indirect objects, and sentential complements. However, it runs into some trouble when producing most complex sentences and has difficulty with the seemingly simple insertion of adjective pre-modifiers. It is possible that the use of better error measures and more extensive training on production could result in significant improvements in these areas.

Analysis of the model's production behavior revealed some interesting characteristics. The model seems to be able to recover quite well from its own production errors. The model tends to make production errors in the middle of sentences, rather than at the beginning or the end. It also seems to make few "state" errors, in which it simply loses track of where it is in the sentence. It also rarely commits errors that involve changing a word to one of a different syntactic class. Most of the errors are either semantic errors, in which a word stem is changed to another one with similar meaning, or borrowings, in which a word is copied from one position in the sentence to a different position. Most of the cases of noun borrowing were between two objects, and usually it is the second object that is produced too early, rather than first object that is repeated. Interestingly, borrowing errors usually retain the number and, in the case of verbs, tense of the correct word for that position. It is often just the word stem that is borrowed. This suggests that these errors occur on a fairly deep level.

Experiment 12 tested the model's susceptibility to producing agreement-attraction errors, as documented in human speakers by Bock and Miller (1991) and Bock and Eberhard (1993). Surprisingly, the model was able to replicate not only the occurrence of agreement-attraction errors, but the fact that such errors are much more common when they involve replacing a singular verb with a less-frequent plural form. A plausible explanation for the fact that both humans and the model are more likely to commit plural agreement-attraction errors is that singular nouns and verbs are treated as a default state, which is overridden by an exceptional plural subject. When a plural noun intervenes between a singular subject and its verb, the verb number is sometimes accidentally co-opted. Because they are the default, singular nouns cause no change in plurality and are less likely to co-opt the verb form.

Experiment 13 evaluated the model's ability to exhibit structural priming if its learning mechanism is enabled. It proved capable of developing structural priming for double objects or prepositional datives that survives over a number of filler items, as found by Bock and Griffin (2000). Furthermore, although priming is strongest in the model when there is a high degree of similarity between primes and targets, it can also result from primes that differ from the target in semantic or surface form.

13.2 Accomplishments of the model

It seems fair to say that, from either a technical or theoretical standpoint, this project was ambitious. The network itself, with 2.9 million links, is considerably larger than other network-based cognitive models of which I am aware. And the range of the empirical data addressed goes well beyond that of other connectionist sentence processing models, which have generally been tailored to address specific areas, such as relative clauses, the MV/RR ambiguity, or structural priming. Importantly, the model incorporates the processes of comprehension, prediction, and production, and in so doing it may be the first implemented model, connectionist or symbolic, to offer an account of the complete link between the surface forms of sentences and their conceptual or message representations.

Few if any connectionist models of comprehension or production have been equipped to deal with complex sentences. Even the prediction models that have been applied to complex sentences were trained on languages that were considerably less complex than Penglish (Elman, 1991; Tabor et al., 1997; Rohde & Plaut, 1999; Christiansen & Chater, 1999b). Although networks have, in the past, been exposed to number agreement, Penglish, because it has multiple verb tenses and passive forms, has a more involved inflectional morphology. Other advances in the Penglish language over those previously used with most connectionist models include the use of determiners, an expanded vocabulary, prepositional phrases, sentential complements, subordinate and coordinate clauses, adverbs and adjectives, and the inclusion of lexical ambiguities. Although Penglish still lacks many important features of English, some of which were listed in Section 5.1.2, I consider Penglish to be no longer a "toy" language. Penglish could be expanded relatively easily to include other important features of natural languages, such as larger vocabularies, questions and commands, non-indexical pronouns, and compound nouns and verbs.

I wish to argue not that the model provides the correct explanation for all phenomena to which it has been applied, or even that it was able to reproduce all of the relevant behavioral data, but simply that its ability to simulate human behavior is sufficiently impressive to warrant further interest in this and other connectionist accounts of human language processing. The model has demonstrated a variety of abilities that are known to be critical for skilled comprehension. It is quite good at processing simple sentences and is able to handle many complex sentences with a reasonable degree of proficiency. It is able to deal with the ambiguities that pervade natural language, from lexical and word-sense ambiguities, to simple temporary ambiguities of auxiliary verbs like *was*, to double object datives, to ambiguous clause boundaries, as in the NP/0 ambiguity.

Tests of the model repeatedly show that, in comprehending both ambiguous and unambiguous sentences, it is sensitive to a variety of factors including biases of structural frequency, verb argument structure preferences, semantic plausibility, and inflectional morphology. Importantly, these sensitivities were not built into the model by design, as in earlier constraint-based accounts (Trueswell & Tanenhaus, 1994; MacDonald et al., 1994a; Spivey & Tanenhaus, 1998). The model's sensitivity to particular statistical factors, and the representations on which they depend, arise naturally from the constraints of its connectionist architecture as it learns to perform the tasks of comprehension and production as best it can.

The model is not without its limitations. For example, it has serious trouble with high-adverb attachments following sentential complements and it can be confused when a sentence contains elements that are semantically interchangeable. However, I would argue that such limitations and susceptibilities to confusion tend to mirror those of human language processors. Limitations such as non-locality costs or syntactic complexity bounds can be placed on other models to try to simulate human performance; however, a key property of the CSCP model is that its limitations, like its capabilities, arise naturally from the processing principles inherent in its architecture.

Some would argue that a neural network model with 2.9 million weights is like a symbolic model or mathematical function with 2.9 million parameters, so of course it is possible to model a finite set of data with so many free parameters. But to treat link weights as free parameters is to miss a very important distinction. The primary danger of too many free parameters is the possibility that the designer has adjusted the model to fit known patterns of data, which, by assumption, will reduce its likelihood of explaining any new data. Such adjustments could occur by hand or through automatic optimization procedures that use the degree of fit to known empirical results as the criterion of success. But neither of these accurately characterizes the way in which the CSCP model was designed or trained. The model does not have explicit parameters that permit the targeted adjustment of its high-level behaviors. Its weights cannot be selectively manipulated to achieve a desired trait. The model's weights were indeed tuned by an automatic optimization process, but the goal of this process, as defined by the error function it seeks to minimize, was not to achieve specific patterns of behavior, only to reduce overall error, either in question answering or in word prediction. The model was not directly encouraged to have a bias toward low-attachment of adverbs, to have more difficulty comprehending passive datives than conjoined transitives, nor to be more susceptible to agreement-attraction errors with plural distractors. These are properties that arose out of the model's architecture and the environment on which it was trained, but they are not directly governed by parameters.

The closest thing in the model to free parameters, in the sense that they are chosen by the designer, are factors such as the size of the various groups in the network, the error measures used, and the learning rate, momentum, and other values affecting its training. While these parameters can have an effect on the model's overall performance, and were selected to try to maximize that performance given the limits of time and computing power, they do not have a direct bearing on the model's more abstract traits. One cannot simply make a certain type of sentence easier or harder relative to another by adjusting the size of the network or its learning rate. Therefore, it would be misleading to argue that a model such as this has an astronomical number of free parameters. There are many different species of parameters that can go into the design of complex cognitive models, and not all are as easily manipulated to achieve desired outcomes. This should be taken into account in our perception of the complexity of a model and our expectation that it will generalize to novel data.

13.3 Problems with the model

The CSCP model is not perfect and there are innumerable ways it which it must be improved and extended to provide a more complete account of human language learning and processing. Major extensions to its scope will be discussed in Section 13.4, while this section considers some of its immediate problems and how they might be resolved.

A number of the required improvements are not to the functioning of the model itself but to the Penglish language on which it was trained and, specifically, to the way in which its semantics are encoded. Encoding sentence meaning using appropriate propositional representations appears to be fairly important to the ultimate behavior of the model as it is quite sensitive to this critical source of information. Two major problems with the semantic encoding were identified while studying the model, both dealing with the ordering of propositions. One was the fact that both actives and passives are encoded subject-first. This places too much syntactic information in the semantic representation and, in practical terms, eliminates the possibility of forming a message with ambiguous voicing for testing syntactic priming. A preferable alternative would be either to encode the thematic roles of a verb in random order or to use a canonical ordering in which agent or experiencer is always first, followed by patient or theme.

The second major problem with the ordering of propositions in Penglish is the fact that the propositions reflecting center embeddings were placed between those encoding the subject and the object of the main clause. This seems to have disrupted the model's ability to encode both the main and the relative clause and resulted in serious problems in the comprehension of such sentences. These problems were not apparent in an earlier version of the model which encoded the verb, its agent, and patient in a single proposition, and thus did not permit the subordinate clause propositions to interrupt those of the main clause. Possibly this can be resolved by simply reordering the propositions to bring those for a single verb in proximity to one another. But perhaps these problems with proposition ordering reflect the fact that ordering of the propositions in the message of a sentence is unnatural. The reason sentence meanings were encoded as a series of propositions was to permit the specification of meanings for arbitrarily complex sentences. There may be a better way to do this that does not require an ordering to be placed on the propositions.

Although the Penglish language has several minor bugs that need to be fixed, such as the frequency of adjective modification of sentence subjects, a more difficult problem to be solved is properly balancing the overall frequency of sentential complements and relative clauses. The frequency of sentential complements was specified in Penglish on a per-verb basis. In order to test the effect of SC-bias on reduced sentential complement ambiguities, Penglish happens to contain a high percentage of SC verbs. As a result, the overall frequency of SCs in the language is unnaturally high relative to that of other structures. In particular, this may have had an effect on the processing of relative clauses, which, in terms of their local surface structures, are quite similar. It may be possible to remedy this only by adding more non-SC verbs to the lexicon to better match the statistics of English.

Another relatively simple and useful addition to the language would be prosodic or punctuation cues to indicate clause boundaries. There is good reason to believe that humans benefit from such markers to help them avoid ambiguities such as the NP/0. If these cues were available, the model would presumably come to rely on them as well. When the cue is absent, it is likely that the model would experience a larger garden-path effect, which may better model human behavior.

One final, and perhaps significant, problem with the method of training and testing the model's comprehension is that it is only asked questions about propositions that are actually in the correct message. Given the sentence "*The boy ate the soup*," the model would be asked questions such as "Who ate?", "What did the boy eat?", and "What was done to the soup?" It would, however, not be asked if the boy used a spoon, if the soup was hot or tasty, or if there happened to be a horse racing by at the time. As a result, the model has pressure to remember the information actually in the message, but it is free to fabricate any other information it likes because it will never be tested on that. To an extent, the ability to infer missing constituents is exactly what we want in a skilled comprehension system (St. John & McClelland, 1988). However, inference of this sort can be a problem when messages are specified for production, where we would prefer that the model not add unnecessary information. Perhaps, inference during comprehension ought to occur at a somewhat higher level of representation than the one tapped in driving production. To discourage any tendency to fabricate information at the message level, it may also be necessary during training to test the model's ability to recognize that a given proposition is not part of the specified message.

We now turn to problems with the model's processing mechanisms themselves. One peculiar finding is the difficulty it experiences when a clause ends in an intransitive verb. As shown in Section 7.3, the model performs better on transitives or on intransitives that are post-modified by an adverb or PP than on sentences that end in an intransitive verb. This suggests that the model is unable to completely finalize its interpretation in the single processing step that occurs following the end-of-sentence marker. It seems likely, therefore, that the difficulty of intransitives is a result of the strict processing limitations placed on the network. The fact that a single processing step is allowed for each word is not intrinsic to the theory, but was a convenient choice for simplicity and computational efficiency.

My theory of human language processing would better be served by a fully recurrent network with truly dynamic processing of variable duration. However, this may be for now too computationally intensive to be practical at this scale. A compromise might be to give the model just two processing steps on each word. Alternatively, and perhaps preferably, we could retain the one-step per word but add an extra step at the end of the sentence to allow the model more time to finalize its interpretation. Not only should this improve the model's performance on sentences for which the end-of-sentence resolves an open ambiguity, it may also better reflect sentence wrap-up effects that are observable in most online reading studies.

Although some interesting similarities were seen between the production errors made by the model and those of human speakers, the model's production abilities overall are not up to the level of its comprehension performance. The model was quite good at producing simple sentences and was able to produce some fairly complex sentences perfectly, but the frequency of production errors for complex sentences was much higher than seems acceptable. It may be that this represents a fundamental limitation of this architecture for producing language. However, before this conclusion is reached some alternatives should be considered.

One alternative is the possibility of using better training methods to improve the model's production performance.

In the current model, sum-squared error was used to provide feedback to the prediction/production layers during training. However, the *divergence* measure is more appropriate for such situations and usually results in improved learning (Rohde & Plaut, 1999). Divergence was not used in the current model because it was unstable early in training and tended to result in an explosion of the magnitude of the link weights. The model's production may improve if divergence can be reintroduced with better limitations to prevent such instabilities.

Production should also improve with an increase in the frequency of production training trials relative to comprehension trials. In the current training method, true production training, with fully clamped semantics, occurred on just 20% of training trials, with 50% comprehension trials, and the remaining 30% somewhere in the middle. Performance may improve if a higher percentage of production trials occurs. This could be weighted toward the end of training to simulate the later onset of production during language acquisition.

Another possibility for improving the training regimen might be to allow joint training of the encoder and comprehension/production systems. Currently, the encoder network is trained first and remains static while the lower half of the network is trained. This was intended to reflect the fact that the encoder/decoder system is meant to represent general cognitive abilities not necessarily tied to language. But it results in the situation that the message representations formed by the network are created solely under pressure to answer comprehension questions, but do not necessarily reflect the needs of the comprehension or production systems. It would, perhaps, be both more accurate and more helpful to permit the encoder system to develop alongside the comprehension/production system. This would allow the message representations to develop in such a way that they are better tuned to the abilities of the comprehension/production system. The drawback of this, aside from the possibility that it may not be a significant help, is that such training would be time consuming and further complicate the model.

Finally, in the measurements of reading time at the disambiguation point following various ambiguities, the model, by and large, demonstrated sensitivity to many of the same factors to which human readers respond. However, one fairly consistent difference between the model's performance and that of humans is that the model does not show drastic increases in reading time when it is garden-pathed. There tend to be small increases, but not the significant pauses that are observed experimentally. There are several reasonable explanations for this.

The first is that the model may not, in fact, be garden-pathed. If the model has received undue exposure to a particular ambiguity, it could actually be better than humans at resolving it. This may be the case with reduced sentential complements, with which the model has received a lot of experience and on which it tends to have low comprehension error rates. Another case of this may be the NP/0 ambiguity. Although human subjects are probably mislead by the lack of a comma when reading such sentences, the model has no experience with commas and thus may be more attuned to the possibility that such an ambiguity is occurring. Another possibility, which is also a case of the model not behaving appropriately, is that it has been so severely garden-pathed that it has no chance of recovering and is in a state of confusion. The model may have, essentially, given up and be waiting until it reaches more familiar ground before making any significant changes to its interpretation. This seems a reasonable explanation for cases in which the model actually read faster in what should have been the hardest condition than it did in supposedly easier conditions. Third, it is possible that the model was not irreparably garden-pathed, but has learned to perform its recovery over several subsequent timesteps, rather than right at the point of disambiguation. This is a reasonable strategy for the model to develop because of the limited processing it can perform on each word. It seems possible that humans will adopt such an extended reanalysis strategy when listening to sentences, where they do not have the option of pausing the input as they do when reading.

There are still other accounts for the lack of strong garden-path effects which would not necessarily imply that the model is performing in a non-human-like manner. One is that the model may, in fact, be restructuring its representations at the disambiguation point, but this is just not reflected in the current simulated reading time (SRT) measure. The choice of components for the SRT was somewhat arbitrary. It is sensitive to changes in representation at the message layer, but it is possible that the model is actually carrying out restructuring primarily through changes at the comprehension gestalt layer, or that restructuring is not linearly related to the change in activation of the units in these layers. This can only be determined by testing the model extensively for other possible indicators of restructuring.

Finally, it is possible that the long reading times seen at or just following disambiguation for human readers reflect true reanalysis. By reanalysis, I mean a process by which the comprehender replays a part of the sentence in order to reparse it, rather than simply restructuring established representations. During normal reading, reanalysis can be affected by actually re-reading part of the sentence. This can be, and is, directly observed in eye-tracking studies. But it is also possible that comprehenders can replay the sentence from a short-term phonological buffer. Such an approach is necessary during masked self-paced reading or auditory comprehension when the sentence cannot simply be re-read. But phonologically-based reanalysis may also occur in eye-tracked reading, where it could simply be faster or more convenient than actual re-reading. Therefore, a lack of regressive eye-movements should not be considered a definitive indicator that reanalysis is not occurring.

Nevertheless, the current model is not intended to account for reanalysis. It is only intended to account for first-pass comprehension. Therefore, if long reading times at disambiguation points actually reflect reanalysis, the model should not be expected to account for them. A more sophisticated model should have the capability of maintaining recent information in a phonological memory and accessing that information when it is deemed necessary. To properly evaluate the current model, or other accounts of first-pass comprehension, we need to distinguish first-pass processing from reanalysis in human subjects. Possibly this can be done through imaging techniques, but it may not be feasible to do this accurately. If that is true, a better accounting of human behavior may require another generation of models capable of full reanalysis.

13.4 Model versus theory

The CSCP model is itself an instantiation of a broader theory of human sentence processing. It would be wrong to think of the model, or at least the implemented model, as equivalent to the theory. Certain properties of the model reflect critical claims of the theory, but others, like the minor details of its architecture and even the specific error measures used in training it, are considered to be largely inconsequential and unlikely to substantially affect its qualitative behavior. Still others, like the lack of prosodic information or any sense of discourse, do have significant behavioral consequences, but do not reflect tenets of the theory, only convenient and temporary simplifications. I will acknowledge that the present theory is not fully fleshed out and some of its commitments remain open. But there are important aspects of the theory that can be articulated at this point. Some of these claims were introduced and explained in Section 6.6; and I will repeat the claims here, but not their explanations:

- The human sentence comprehension and production systems can be effectively modeled by a connectionist network.
- The comprehension and production systems are highly integrated and rely on common representations.
- The sentence processing system does not contain clear modules, but it does have functional specialization.
- In comprehension and production, syntactic processing is not clearly segregated from lexical or semantic information.
- The comprehension system is constantly engaged in formulating covert predictions.
- Learning to predict accurately during comprehension, in the presence of varied levels of advance knowledge of the message being conveyed, is a primary mechanism by which we learn to produce sentences.
- Production is regulated through self-monitoring.
- Production proceeds both incrementally and globally.
- Learning to comprehend language is largely based on an ability, either learned or innate, to extract meaning from observation of the world.

The following are some additional observations and qualifications to the theory:

- Critical aspects of the theory include the fact that the model is composed of a large number of processing units with non-linear response properties, that these units are interconnected in a relatively dense manner that does not place tight constraints on the specific internal representations that can develop, that a substantial portion of the connections are recurrent, and that the initial weights on these connections are random, or at least unstructured.
- It is not critical to the theory that the model be simple recurrent, that it include self-recurrent connections, or that there is a lack of noise in the system. Ultimately, a fully recurrent network will be a more appropriate way to instantiate the theory and may obviate the need to define simulated reading time measures. However, an important constraint in determining the nature of the representations formed by the model is that there is pressure to engage

in minimal processing per word. Processing time is strictly bounded in simple recurrent networks, but can be more loosely constrained in fully recurrent networks.

- I wish to argue that it is not critical to the theory that the model learn via backpropagation and particularly not via backpropagation through time. However, I suspect that the use of another training method would result in some qualitatively different behaviors than the current model is likely to express. For the time being, this is something of a moot point since I do not believe we currently know of a training method other than backpropagation capable of enabling a multi-layer, recurrent network to learn a complex task like sentence comprehension.
- To return to the issue of modularity, I have essentially separated the CSCP model into two systems, the encoder/decoder system and the comprehension/production system, which interface at the message representation. This was mainly done to simplify the model for explanatory convenience. One might be tempted to call these clear cases of modules, and I would be hard-pressed to argue that they do not have modular aspects. But, importantly, the theory does not make any claims about the functional significance of informational encapsulation between these systems. Semantic information certainly plays an important role in both systems, and the semantic representations developed by the encoder system determine, through training, the properties of the comprehension/production system. If joint training of these systems were to be added to the model, as it eventually should be, the distinction would break down further and syntactic information would come to reside in the encoder as well.
- The theory assumes that semantic information in the form of distributed encodings of propositions is available to the learner from a source external to the language system. However, it does not claim that these propositional representations cannot also be shaped by experience gained through the use of language.
- The theory makes the claim that syntactic information need not be built into the model. It does not deny that syntactic-level representations exist in the trained model. The claim is that syntactic-level abstractions are useful generalizations that can be learned through exposure to the environment. However, the theory makes the further claim that there is no level of representation that is purely syntactic. Semantics plays a role from the recognition of words in the acoustic or visual signal right through to the extraction of a message.

A lesson to be learned from the current model, to the extent that it is accepted as a valid account of human sentence processing, is that the difficulty of comprehending a sentence is a function of both its semantic and syntactic properties. In searching for explanations of sentence complexity, we may tend to restrict our explanations to syntactic factors, such as the frequency of structures or their difficulty as a function of locality or number of attachments. These are important. But equally important are semantic complexity factors, such as the plausibility, and/or frequency, of propositional relationships and the degree to which elements of the sentence can be confused with one another. Some sentences may be hard to comprehend not because they have complex syntax but because they have complex semantic relationships. The model leads one to predict that we rely to a considerable extent on semantic constraints in interpreting sentences, particularly when syntax is difficult, and that we will easily be confused when semantic plausibility goes against the correct interpretation as determined by syntax, but will be able to overcome difficult syntax when the correct interpretation aligns with semantic biases. A complete account of sentence comprehensibility must explain not just syntax, but the role of semantic complexity as well.

There are many possible avenues for further elaboration of the model. Along with the claims of the theory, Section 6.6 discussed some of the significant limitations of the model's current scope and how these might be addressed in future work. Important among these, though not easy to approach, is providing a more reasonable account of how semantic information is extracted from the world or the discourse context to serve as a target for learning comprehension. Another will be a better account for how we learn new words and incorporate them into our existing language. The current assumption, that the model is exposed to all words from birth, is clearly over-simplified. This brings up a more general point, which is the need to provide a full account of language acquisition. An important property of the CSCP model is that it is capable of learning to process language from exposure to the environment, without relying on extensive innate syntactic knowledge. However, the model is not yet intended to directly account for the learning mechanisms responsible for this, nor to reflect the full progression of language acquisition. It remains to be seen if the model can account for the bounds on fluency achievable by most late second-language learners.

One final issue to consider is the relationship between the model and the environment, or language, on which it is trained. Although the behavior of the model is, to a large extent, a function of this environment, the environment itself is logically separable from the model and the computations and principles that define it. The environment should not be manipulated arbitrarily to achieve a particular result. In order to model the behaviors of English speakers, the environment should be constrained by the known properties of English along any dimensions that are believed to be relevant to such behaviors. Ultimately, it will be necessary to distinguish the properties exhibited by the model in specific linguistic environments from those likely to be exhibited by it in any environment, or at least any environment fulfilling certain requirements of a natural language. In this approach, the current environment is treated as a given. The theory does not, for the moment, seek to account for the fact that language is the way it is, only how a new initiate can learn to process a given language. Ultimately, if a descendant of this theory is correct, we may find that the abilities and limitations of its models recommend them for learning only certain types of languages, but that these languages share the universal similarities of all human languages. But such is optimism.

13.5 Properties, principles, processes

Demonstrating that the model can replicate known patterns of data is critical if it is to be taken seriously as a potential account of human sentence processing. But it is also important that we be able to explain at a more abstract level how the model and, by implication, the human brain operate as they do. This will give us the tools to reason about and predict the behavior of the model, as well as the effect of altering it or exposing it to a different environment. There is, however, a tendency to expect such explanations in the form of explicit representations and rules—that this is the only form in which we can obtain true understanding because it is the easiest for us to reason about. But it is possible, and it is a belief shared by many connectionists, that complex mental abilities, with language among them, cannot be adequately described at an algorithmic level expressible in terms of rules and symbols. The observable properties of the behavior of a natural system, as well as the intermediate-level principles that help to explain them, arise from the implementational details of its basic processes. The intermediate-level principles are important, but they will not be sufficient to fully explain its behavior. By starting at just this level of description it is unlikely that we will be able to induce our way down to the underlying mechanisms. The level of neurons and synapses, or their idealizations in units and links, should be a concern not just of neuroscientists but of cognitive scientists as well, lest we be chemists attempting to understand the behavior of elements and compounds with no theory of atoms or electrons from which to reason.

The relationship between the implementation and the behavior of the CSCP model will be discussed at greater length in the remainder of this section. For the time being, suffice it to say that we should not seek to understand its functioning necessarily in terms of rules and parameters. In reasoning about the model, we must consider the pressures of the tasks on which it is trained and the mechanisms available to it for representing and manipulating information, and expect answers to be in terms of the interaction among general principles of information processing in such systems which may, at times, lie in opposition to one another.

The CSCP model is conceptually decomposable into two main components, the semantic encoder/decoder and the comprehension/production system. Due to the nature of their tasks, each of these has its own abilities and limitations which explain different aspects of overall human language processing behavior. Some of the observable properties of the encoding system include its sensitivity to frequency, its tendency to confuse semantically similar constituents, and the "U-shaped" profile of its ability to recall a series of propositions. The comprehension system has somewhat different traits, due to the different nature of its tasks, but it seems to share sensitivity to frequency and a tendency to confuse similar constituents. In this case, however, similarity may be based more on syntactic rather than on purely semantic properties. The comprehension system is also marked by a sensitivity to locality effects and an apparent ability to resolve ambiguities through the use of limited parallelism. These and other properties will be discussed further. For the time being, let us note that both systems perform their jobs through reliance on a variety of weak constraints that provide hints at the proper response or interpretation. Although these can be described in such terms as semantic plausibility, pragmatics, or structural biases, they all have correlates in the co-occurrence statistics of the linguistic environment.

13.5.1 Sensitivity to statistics

A popular view of connectionist networks is that they are merely general statistical approximators, and this is seen as a limitation to their ability to explain human language processing. I can think of three ways to respond to this. The first is to point out that, yes, such networks are quite sensitive to the statistics of their environment, and that this simple property happens to enable them to explain a large portion of observed human behavior. Jurafsky (1996), for example, accounts for a range of psycholinguistic data with a model that rests directly on observed frequencies, although his model is an abstract one not described at the level of interacting units. For many connectionists, frequency is the default hypothesis because we know our models, and people, are sensitive to it. Therefore, if frequency biases can explain the phenomena of interest, there is no need to resort to alternative explanations. There is nothing the theorist should like better than a single account that explains a lot of data.

However, the second response is that making use of statistical information in the environment is, in practice, not so easy, due in part to what has been termed the *grain problem*. There are an infinite variety of frequencies to which a learner could be sensitive. If the learner tries to pay attention to too many of them, it would seemingly be overwhelmed by the enormous task of cataloging all of the information extractable from each sentence encountered. But if the learner fails to pay attention to a useful statistic, it will never be able to use that information to its advantage. Furthermore, most of the interesting statistics require generalization based on higher-order representations of the input. The alternative is to be sensitive only to the relationship between observable surface elements. A trigram model, for example, bases its knowledge purely on the co-occurrence of words, and is a notoriously impoverished description of natural language. Of course, even this ability requires generalization with spoken input because identifying words in an auditory signal requires abstracting over variances in the acoustic signature of words across speakers and contexts.

Adding to the problem is the fact that statistics at different levels may not agree and may pull the system in opposite directions. For example, consider examples (71a) and (71b) below, which illustrate a reduced sentential complement ambiguity. Neither of these sentences seems particularly difficult to comprehend. What would a frequency-based system predict to be the preferred interpretation in the two ambiguous regions? If the model goes by the overall frequency of NP versus sentential complements, the NP reading should be the preferred one, since, according to the Penn Treebank, the overall probability of an NP complement is 4 to 5 times that of an SC. However, if we only consider occurrences of the verb *concluded* (or should we include its morphological variants as well?) the ratio of SCs to NPs is 5:1, suggesting an SC bias. But if we only count the reduced SCs, there are slightly fewer of these than there are NPs. Added to this is the fact that the semantics of the subject clearly affects the possible senses of the verb and thus the likelihood of an SC. A pungent odor is unlikely to be drawing conclusions. And perhaps the prediction should not be based on a general semantic property of the verb's subject, like its cognitive capacity, but on some more specific property or on the individual lexical item itself.

(71a) The landlord concluded the pungent odor was my fault.

(71b) The pungent odor concluded my list of complaints.

Thus, any statistically based model, particularly one that has access to many levels of description, must also have a facility for combining all of this information in formulating its predictions or hypotheses. Incidentally, even trigram models encounter this problem. Generally, they will also track bigram and unigram frequencies. The interesting difference between trigram models is in the degree to which they rely on one of these sources over another. The trigram has the advantage of being more specific to the current situation, but the bigram and unigram are based on more data and thus may be more reliable.

Any reasonable frequency-based model of human language must, therefore, have two basic properties. One is access to higher-order principles, such as the concept of a noun-phrase, a clause, alternative argument structures, or animacy, on which to base the statistics it gathers. The other is a mechanism for combining these statistics in making a decision. Higher order concepts, the associated statistical information, and the procedures for using this information are built into the design of models like Jurafsky's (1996) or the Spivey and Tanenhaus (1998) interactive activation model of ambiguity resolution. This is fine as an initial demonstration of the usefulness of statistics and for addressing certain issues. But unless one thinks such concepts are innate, a complete theory must explain how the model develops them in the first place. The burden for nativists is equally great, for they must explain how the concepts were developed by evolution. Creationist nativists may relax.

As I argued in the introduction, one of the most intriguing properties of connectionist networks is their apparent ability to develop higher-order representations to facilitate the mapping from their inputs to their outputs. This has been demonstrated in one form by the fact that Elman's (1990) prediction network, when trained on words with arbitrary input representations, developed word representations at the hidden layer that clustered into syntacticosemantic classes representing the distinction between nouns and verbs, animate and inanimate nouns, and so forth. As shown in the current work, connectionist models are also sensitive to statistical information at various levels, such as biases of verb argument structures and the semantic relationships between verbs and their subjects and objects.

I am not arguing that connectionist backpropagation models are necessarily optimal in their ability to form useful generalizations and solve the grain problem. But they do at least appear to be capable of offering one solution to these problems. The ultimate question will be whether their limitations in this regard correspond to the limitations of the human ability to make use of statistical information in the environment. What is not clear at this point is exactly how the model develops higher-order concepts, how this relates to the model's observation of statistical information, and how multiple sources of information are combined. Together, these can be summarized as how the model learns. Studying these processes in detail, perhaps in simpler and better controlled models, will be an important area of future research and should contribute greatly to our current understanding of information processing in connectionist networks, which remains in its infancy.

What does seem apparent is that critical factors in determining how the model develops and uses abstract concepts based on the statistics of the environment are the nature and extent of the resources available to it for representing and manipulating information. Resource constraints determine the amount of information the model can store and the complexity of the representations it forms. A network with severely limited resources will be forced to form the most general and most useful abstractions. For example, such a model might only be sensitive to the overall frequency of words in the language. Or, at a lesser extreme, the overall frequency of possible argument structures across the verbs. If given more resources, in the form of units and connections, it may be able to distinguish different classes of verbs, and so on. Although the general properties of the mechanism by which the current model learns to process language is an intrinsic feature that is shared across all versions of the model, a network with more or fewer units in key layers may exhibit different sensitivities to various forms of statistical information. Therefore, the third response to the suggestion that neural networks are merely sensitive to statistics in the environment is that the manner in which a network solves the grain problem is determined by the form in which it is constrained to process information.

13.5.2 Information processing with distributed representations

To provide a more complete explanation of the model's behavior, then, we must begin to consider the information processing resources available to it. It is not practical to reason about the model's behavior purely at the level of individual units and links. Rather, a more useful abstraction is to think of the model as representing information by means of vectors, or points, in a high-dimensional space. Let us define the model's state to be the pattern of activations over all of its units at a particular point in time. These activations could be arranged as elements of a vector in a high-dimensional space. We can then think of information processing in the network in terms of the representations that form in that space and the transformations between them from one time step to the next. Information processing in such spaces is not well understood, but I will attempt to offer some basic insights into how we might reason about it.

Representational constraints

The first thing to note is that the representational capacity of the space will be determined by the number of dimensions it has, the range of values that can occur for each element of the vector, and the noisiness of those values. The first of these needs little explanation. All other things being equal, more dimensions allow more possible vectors and will permit more information to be stored. But the range of values allowed along these dimensions is also relevant. A binary vector with just two possible values per dimension can have only a finite number of distinct states. A vector with discrete elements that can take on more that two values will have a larger, but still finite capacity. But once we permit the dimensions to take on real values, there is, in theory, the potential for infinite representational capacity. A single real-valued number, even within a finite range, that has no bound on its precision could store an arbitrary amount of information. However, if the values in the vector are noisy, the representational capacity will diminish. The number of distinguishable states will be a function of the noise level and our willingness to accept occasional errors.

When we talk about representing information in high-dimensional space, it is useful, although not necessarily accurate, to limit our thinking to the storage of a set of discrete pieces of information. For lack of a better term, I'll refer to these pieces of information as *chunks*, without intending to adopt any theoretical commitments that might be associated with that word from its typical use in cognitive science. Let us assume that the model is under pressure to remember a large number of chunks, perhaps as many as possible. There are a variety of ways in which multiple chunks of information could be stored in a high-dimensional vector space. Perhaps the most straightforward is to assign each dimension of the vector to a particular type of chunk. The presence or absence of the chunk is encoded by the value along the single dimension associated with its type. This is essentially what we mean by a localist representation. It has the advantage that it is easy to understand and manipulate, but a serious flaw in the fact that it limits the total number of different types of chunks that can be represented to the dimensionality of the vector.

What if we want to be able to encode a large number of different types of chunks, many more than the dimensionality of the space? An alternative approach is to use a distributed vector to represent each type of chunk. In the simplest case, we could associate each one with a moderately sparse random vector. Importantly, a single dimension of the vector would now be involved in representing many or all different chunks. Multiple chunks can be stored by summing their vectors, and perhaps normalizing the result in some way if there is a constraint on the magnitude of the permitted values. The question then is, how do we later determine exactly which chunks have been stored. One way to do this is to calculate the dot product of the representation for that chunk with the full vector. If the chunk has been stored in the vector, this dot product is likely to be large and if it surpasses some threshold we will consider the chunk to have been stored. In theory, such a technique permits the storage of a large range of different types of chunks. But there is now the possibility that the memory will be lossy. It is possible that the dot product for a chunk that has been stored may fall below the threshold due to the normalization step and interference from the vectors for other, overlapping chunks. There is also the chance that a chunk will be falsely remembered because its representation happens to be similar to the one formed from the combination of the chunks actually stored. But, in practice, such techniques based on the superposition of distributed representations can make relatively efficient use of a high-dimensional space.

Now, imagine that some chunks tend to be stored more often than others. We can improve the average recall performance of the system by making sure that the common chunks cause minimal interference. This is done by assigning the common chunks representations that overlap as little as possible with all of the others. A simple approach might be to assign each of the top n chunks its own dimension. The remainder of the chunks will use random distributed representations across the other dimensions. The advantage of this is that the n most common chunks cause little or no interference with one another or the less common ones. Overall, if the common chunks are common enough, such a system could have better recall performance than one which ignored frequency and treated all the chunks the same. One could imagine, however, that a better solution would be somewhat more flexible and would adjust the density of the representations and the selective use of certain dimensions based on the relative frequencies of the chunks.

This begins to explain how frequency can come to play a role in a system with a memory consisting of a high-dimensional vector space. Given a limited representational space, and the requirement that the system store a large amount of information in the form of distinct chunks, a good solution is to assign the most common chunks the smallest representations, possibly using a reserved part of the space, that result in the least interference. The model will effectively have a larger memory capacity for frequent chunks than for infrequent ones. This approach is clearly related to issues of compression in information theory. Given a bounded capacity channel and the requirement that as much information as possible be transmitted in a short amount of time, compression systems will universally assign shorter representations to more frequent pieces of information. The distinction between that case and the idea of storing information in a bounded capacity space is mainly one of temporal versus spatial representation. To those familiar with compression, it is a very useful metaphor for reasoning about the operation of multi-layer neural networks.

Procedural constraints

A recurrent network processes information over time by transitioning from one state to the next, which can be thought of as moving through state space. The new state is a function of the previous state and of the external inputs, which could be considered part of the overall state if one wished. Aside from the sheer representational capacity of the space, there is an additional set of constraints deriving from the model's ability to make transitions from one state to another. The CSCP model is quite limited in the type of state transitions it can perform. The units are defined such that each component of the new state is a linear combination of components of the previous state, passed through a sigmoidal squashing function. Furthermore, the limited patterns of connectivity between units mean that only part of the previous state is considered when computing the new value for an element.

Therefore, there is a limit on the complexity of the transitions that the model can effect in moving from one state to another. A simple recurrent network is particularly limited because it is forced to undergo just one state transition between inputs. Fully recurrent networks, have the advantage that they generally undergo multiple processing steps in moving from one stable state to another. However, their power is often restricted by requiring that each transitional state be a weighted average of the previous state and the desired new state. This forces the movements through state space to be relatively continuous and places a strong bound on their complexity.

Because of these procedural constraints, connectionist models are forced to structure their representational space so as to make the necessary state transitions feasible. The failure to do this may be one reason that the current model has trouble when a sentence ends in an intransitive verb that is optionally transitive. The model must go from a state in which either reading is possible, to one in which the intransitive reading has been committed to, which it is apparently unable to do sufficiently.

Earlier we discussed how a set of information chunks might be stored in a high-dimensional space. Now that we are talking about recurrent networks and the processing of information over time, we must consider a somewhat different set of requirements. On each time step, new information comes into the system and it must produce outputs appropriate for the given task. Therefore, on each step the model must re-represent the information present in its previous state, possibly discarding information deemed no longer useful, and incorporate the new information. Unless information is to be discarded, the model may need to make room for the new information or risk interference and information loss. One way it can do this is by compressing the old information. Of course, the complexity of the transformations possible in a single time step, and thus the degree and type of compression achieved, will depend on the processing limits of the connectionist architecture.

Again, the issue of frequency is relevant. Under pressure to store as much information as it can, frequent chunks of information should be compressed more tightly. Presumably, achieving higher compression is more difficult and the model will have to commit more of its procedural resources to operating on high frequency information. Therefore, a system such as this has an interesting property. High frequency chunks of information will tend to consume less of the representational resources, but more of the procedural resources. Although the complexity of the representational transformations that can take place on a single time step are limited, over several iterations of this process, some relatively complex transformations can take place.

For example, Tabor (1998b, 1998a) illustrated how a recurrent network could recognize a context-free language by simulating a stack-based memory using two infinite-precision real values to encode the state of the stack. The stack symbols are represented as points, or vectors, in this two-dimensional space. As each new value is placed on the stack, the previous value is scaled so that it occupies only half its original space and then added to the vector for the new symbol. As a result, the points in the space that represent valid stack states form a fractal. With three stack symbols, they can be arranged in the Sierpinski triangle fractal. Importantly, this process has the property that the information that was first placed in the stack gradually begins to occupy an ever shrinking and less accessible portion of the space. The only practical way of getting at this information is to reverse the encoding process and remove the stack symbols one at a time. Thus, complex representations can be formed by simple processes, but only by iterative application of those processes.

This brings us to the issue of accessibility. Information that will be needed in the short term must be accessible, in the sense that the model can make immediate use of it in computing its next output or internal state. Information that will not be needed for a while, as in the case of the first elements placed on Tabor's stack, can be more highly compressed, which frees space for new information, but with the drawback that the old information cannot be immediately accessed. There is also the possibility of some loss of information in the compression and re-expansion of the old information. The fact that recurrent connectionist networks will operate in this way, given the representational and processing constraints that they face, and the requirement that they perform the given task as best they can, explains the emergence of locality effects in these models. It is relatively easy if new information must be integrated with recent information because the recent information should be relatively accessible. But the model will have more difficulty if new information must be integrated with old information, because the old information may be in a form that is inaccessible, due to complex iterative transformations, or even lost altogether.

In order to account for the U-shaped pattern of recall exhibited by the message encoder system in the CSCP model, one could speculate that some other factors are at work that account for primacy. When the first proposition arrives, the model is in a consistent initial state and there is no other information to worry about. Therefore, encoding that proposition in the message representation is relatively easy. It is made even easier by the fact that the first proposition nearly always describes the subject of the main clause, and the model therefore even knows what to expect. As a result, there will be little error in storing this proposition and it may even take up an undue amount of the representational space. Therefore, the worst recall performance should be expected on the second or third proposition stored.

Now, the current model is doing a lot more than just storing and recalling arbitrary chunks of information. The encoder/decoder network is basically a sequential memory device, but the chunks of information that are given to it, in the form of propositions, come with their own internal structure. Certain components of the propositions are more similar than others. The actions tend to share features with one another, as do the objects, and certain objects are distinguished only by a single bit. The model can make use of this structure to aid compression of the information in the proposition. To give a simple illustration, all humans are animate, so the model need not store both facts. It is sufficient to store human and the animacy can be reconstructed later. Likewise, if the middle element of a proposition is a thematic role, the left element must be an action, so any properties common to all actions need not be explicitly stored. Frequency is still relevant, but we can now think of frequency not just of chunks as a whole but of portions of their representations. Frequent features, or pairs of features that tend to co-occur will be more easily compressed, making available additional room for rare features.

The introduction of a similarity structure does have a downside, however. Chunks of information with arbitrary encodings can be given maximally distinguishable representations, but chunks of information with similar encodings are more easily confused. This gives rise to another property of the model, the greater tendency for error when two similar pieces of information are stored. This may be the main factor that leads the production system to commit borrowing errors. For example, in place of one noun it sometimes produces another one that should occur later in the sentence. It is more likely to do this when the nouns share similar semantic or syntactic roles, such as two objects. The tendency to confuse similar things is also evident during comprehension when a sentence with two prepositional phrases or two adjectives tends to result in higher error rates than might be expected.

Although the message encoder and decoder is basically a sequential memory, the comprehension and production system is doing something that is conceptually far more complex. While the message encoder is given all of the information that it must store, the inputs to the comprehension system are sequences of phonological representations with no systematic mapping to syntax or semantics short of information that can be gleaned from inflectional morphology. The outputs of the production system are even more impoverished. In order to perform these tasks with any degree of proficiency, the network must re-represent these inputs, or the information destined for output, in terms of accessible and functionally relevant features. Thus, the model must develop its own structured representations. If two words are functionally similar, such as two prepositions, it is in the model's interest to give them similar representations, such as a shared "preposition" component. Then, any processing steps that are common to the two words, such as the prediction of an NP following them, do not need to be implemented redundantly. They can be implemented once based on the shared prepositional component. Deriving a higher order feature such as this does two things. It increases the similarity of the two prepositions and creates a higher frequency feature, which makes compression easier and has the effect of reducing demand on the representational resources of the model. It also makes it possible for the model to consolidate functions common to the prepositions, which reduces the demand on the model's processing resources.

Thus, the development of higher order representations can be seen as a solution to the problem of performing a complex task optimally given limited representational and procedural resources. It also accounts for the model's ability to generalize. If processing depends on features shared among a class of representations, then any learning that takes place on one member of the class will tend to transfer to the other members. In this example, I have used the case of two prepositions, but these arguments can be applied to more abstract syntactic or semantic elements as well. But sharing of representations between chunks does involve a tradeoff. If generalization goes too far and individual characteristics of the chunks are lost, the model will be unable to solve any task for which the two are not functionally identical. Whether the model commits to a generalization or maintains distinct properties will depend on the relative usefulness of maintaining the distinction to that of reducing the required resources to make room for other information. The advantage of using distributed high-dimensional representations is that the decision need not be a discrete one. The network can gradually adjust the representations of two seemingly equivalent chunks until they are nearly similar. But if the network then discovers that there is a useful distinction, the representations can be gradually pulled apart again, perhaps retaining a shared core. The dynamics of processes such as these in connectionist models based on Kohonen's (1984) self-organizing map have been explored by McClelland and Thomas (1996).

The concepts of locality, accessibility, and information sharing also offer an explanation for why it is so confusing to the model when the propositions related to the relative clause intercede between those encoding the subject and the object of the main clause in the meaning of a center-embedded relative clause. The encoder system is accustomed to encountering the subject and object propositions of a transitive clause in immediate succession. The model can more efficiently represent this pair of propositions by capitalizing on their mutual information, such as the fact that they both refer to the same action, which need not be stored twice in the message representation. When the second of the two propositions arrives, it can be easily incorporated with the first because the first is recent and accessible. But problems arise if the two propositions are separated by those of the relative clause. The model may be so used to incorporating the first two propositions that it tries to incorporate the subject of the main clause with the subject or object of the embedded clause. Even if it avoids this mistake, it may have trouble incorporating the object of the main clause with its subject because the subject is no longer accessible in its accustomed form.

The internal representations used by the model during parsing remain poorly understood. But the fact that it is able to successfully recover from many temporary ambiguities, coupled with the processing constraints placed on it by virtue of the fact that it is a simple recurrent network permitted just one processing step per word, suggests that the model's ability to resolve ambiguities must rest on at least a limited degree of parallelism. The frequent and extensive reanalysis required in a serial parser is not compatible with these constraints. The nature of the model's ability to represent possible syntactic interpretations in parallel can also be understood in terms of information processing based on high-dimensional representational spaces. The model has an incentive to maintain as many possible interpretations as it can to avoid being stuck on a garden-path. But there is a limited space in which to represent these possible structures, so some may be lost altogether and all must be compressed to some extent. But by devoting more of the representational space to interpretations that are likely in the given context, the model will keep them more accessible and will, on average, encounter the least difficulty when the ambiguity is resolved. Therefore, the model can be reasonably thought of as a ranked, parallel parser (Gibson, 1991).

Some readers may have noted an apparent contradiction in the principles I have evoked in talking about the model's use of its representational space. Earlier, I claimed that more frequent chunks of information would be assigned smaller representations, using fewer resources. Now I have argued that more likely structures will be given *more* representational space. This is not, inherently, a contradiction. The likelihood of an interpretation, as I use the term here, is specific to the sentence currently at hand. By frequency I mean the overall rate at which something occurs in the experience of the model. Frequent structures are often likely. Although likely structures are given more representational space so that they are more accessible, frequent structures will, through training, be assigned more concise representations. Thus, even when those structures are the most likely ones, they will require less of the capacity of the network, leaving more room for other possible structures. For example, the transitive and sentential complement argument structures both occur quite frequently. Therefore, they will require fewer resources to encode them, which makes it easier for the model to store both interpretations simultaneously in accessible form. This is one reason why reduced sentential complements are a relatively easy ambiguity to resolve.

One final consequence of the model's representational and processing limitations is the need to prepare in advance for upcoming structures. If the system is purely reactive to its inputs, an input that requires a complex change in the model's state will present a problem. To give a simple example, imagine that the model has encountered a subject and an optionally transitive verb and has lazily filled its entire representational space to encode them. When the object arrives, the model must compress the information already in its memory at the same time that it incorporates the new constituent, which is a relatively difficult state transition. If the model had earlier capitalized on the fact that an object was likely, it could have performed some of this process in advance, making room for the expected object. Such preparations for expected future structure can account for certain points of comprehension difficulty, as evidenced by slower reading. For example, if the model encounters a conjunction following the main clause, it should immediately know that it must prepare for a whole new clause and this will result in the initiation of a significant restructuring of its state.

13.6 Conclusion

Chomsky (1965, 1975) has proposed the influential idea of a *competence/performance* distinction in language processing. Competence is viewed as the true linguistic knowledge possessed by a fluent speaker of a language, where the term knowledge encompasses all principles, procedures, and rules involved in normal language processing. Performance, on the other hand, involves limitations in our actual ability to use language in practice, including the effects of working memory limitations, distractions, and so forth. The implication is that it is useful, even preferable, to try to explain human language competence in isolation from memory limitations and other performance factors. However, as we have found in discussing the properties and principles of the CSCP model, working memory in recurrent connectionist networks is not a basic commodity that can be expanded, as in a digital camera. Memory is intrinsic to the functioning of the system and cannot be distinguished from the representations used by it or its procedures for operating on those representations (MacDonald & Christiansen, in press). To accept "competence" as a valid construct is not simply to adopt a definitional framework from which to approach the study of language, it is to make a mechanistic claim about the possible architecture of the human language system.

I have characterized the behavior of the CSCP model in terms of a set of properties, including a sensitivity to and advantage for processing frequent structures, a propensity for developing higher order representations to support generalization, a preference for locality in integrating information arriving at different times, a tendency to be confused when two similar things must be maintained in memory, the need to prepare in advance for expected structure, and parsing of ambiguous structures that is rooted in ranked or graded parallelism. Making reference to these properties is a useful way to describe the model and reason about its behavior. But I have also tried to illustrate, albeit with minimal empirical or theoretical support and extensive waving of the hands, how these properties can be seen to emerge as a consequence of more basic principles operating in the model. These principles can be expressed generally as the demands of performing complex sequential tasks with a limited representational space in the form of a high-dimensional vector and with a limited capacity to transform those representations. A recurring theme is that we can reason about the model as if it were, among other things, attempting to compress information to maximize what it can store and bring to bear in solving the given task, while keeping the information that is most likely to be useful maximally accessible.

Sensitivity to frequency derives from the fact that the model must fit as much information as it can into the available space and more resources will be devoted to efficiently and accurately compressing and decompressing frequent information, ideally resulting in less interference with other stored information. Locality derives from the fact that successive re-representations of information, involving compression and possible information loss, make older information less accessible, with the caveat that primacy may also play a role. The development of higher-order shared representations enables more efficient use of representational and processing resources and leads to generalization. But shared information, and representational similarity in general, can result in interference and confusion. While these principles and properties, and others like them, are a useful way to think about the current model, they do not enable us to fully characterize or predict its behavior. To do this will require deeper analyses and more explicit connections to the model's architecture, the actual representations it forms over the course of sentence processing, and the specific computational properties of the primitive elements upon which its operation rests.

In comparison to this form of explanation, it is interesting to consider Gibson's Dependency Locality Theory (DLT) of online sentence complexity (Gibson, 2000). The DLT holds that the complexity of a sentence at a given word rests on two factors. The first is locality, which is operationalized as the number of discourse referents falling between the word and the phrasal head with which it must be integrated. The second factor is referred to in the theory as memory cost, but is specific to the memory required to represent the expected upcoming structures necessary to complete the sentence, such as the verb phrase following a subject. I have argued that analogous factors are at work in the current model. Locality in the CSCP model was discussed at length, with some differences being the fact that the model's locality measure may also include primacy effects and could be sensitive to frequency factors that also bear on accessibility. Gibson's memory cost is related to the idea that the current model must prepare for expected structure in advance to reduce the computational burden at later points. Therefore, the DLT and the current model are largely compatible and could be considered different levels of description of the same system. The DLT has certain advantages, including that it is relatively well-defined and is uncommitted to particular mechanisms, and therefore makes clearer predictions. But the DLT merely stipulates the importance of these two factors. The CSCP model has the advantage that it provides a demonstration and an explanation for how they can arise from more basic principles at work in the model.

This is a good example of the potential relationship between connectionist approaches to understanding the mind and alternative levels of description and explanation. A key tenet of what I consider to be strong connectionism is the belief that complex, high-level behaviors arise from the details of information processing in the low-level, neural substrate of the brain. Certain behavioral traits arise in a direct and clear way from the implementational level, but others emerge via more circuitous routes. The fact that the CSCP model could replicate the higher rate of plural agreement-attraction errors, despite the lower frequency of plural verbs, was not an obvious outcome, nor one that I even yet fully understand. Importantly, connectionism need not deny the usefulness of higher-level descriptions or explanations of behavior, such as the principle of locality and the cost of memory usage, which make it possible to understand and reason about a complex system. But a collection of such principles is unlikely to fully characterize the behavior of the complex system, nor to address phenomena that derive from deeper explanatory or functional levels.

This work has been one attempt to demonstrate how a theory of sentence processing conceived of and implemented as a connectionist network can explain a broad range of human language behavior, as well as to provide an account of the emergence of intermediate-level explanatory principles from the network's architectural constraints.

Appendices

Appendix A

LENS: The Light, Efficient Network Simulator

This appendix briefly describes the LENS neural network simulator, which I wrote in the course of my graduate research on neural networks and which was used in running the CSCP model. In order to implement a network of that size, it is necessary to have a simulator that is both memory and time efficient, but also sufficiently flexible to allow easy modification of and experimentation with the network, as well as one that provides graphical interfaces to facilitate interaction with the model. My experience with some other simulators early in my graduate studies led me to believe that they were not up to the task of the projects I envisioned at the time and that considerable improvement was possible.

Neural network simulators tend to be of two basic varieties: very simple, fast programs designed for a specific type of network on the one hand, and large, graphically intensive systems targeted at diverse users on the other. Although generally quite fast, the former suffer primarily from inflexibility. They are not easily extended to new network architectures. This increases development time for new simulations and prevents multiple users from sharing a single platform, which hinders collaboration and verification of results. Without a sufficient command language, simple simulators are typically limited to a few command-line instructions and must be changed at the source code level and recompiled to perform more complex experiments. Without adequate visualization tools, understanding and debugging a network can be very difficult. A final drawback of simple simulators is that they are often less accessible to non-programmers and are thus not good platforms for introductory courses or for broad publication of methods.

At the other end of the spectrum, large-scale systems attempt to provide enough flexibility to satisfy most users' needs. However, it is quite impossible to anticipate every feature that might be desired and, ultimately, sophisticated users will need to modify the source code of even the most comprehensive programs. This is not easy in most complex software. Even if the code can be understood and modified, without a convenient method of dissociating the original code from a user's changes, those changes will have to be reapplied by hand to any new releases.

LENS was designed to fill a middle ground between large and small simulators, with three primary goals in mind:

1. **Speed**: The development of ever faster machines does not reduce the need for an efficient simulator. The networks we attempt to simulate will grow in pace with the available resources. Speed results from fast inner loops and conservative memory use, with particular attention to cache performance.

2. **Flexibility**: A scripting language and large command set allow most experiments to be performed without the need to modify source code. Unit behaviors can be composed from several input, transfer, integration, and noise functions, providing a wide variety of standard unit types.

3. **Customizability**: When it is necessary to make modifications to the source, it can be done with minimal interaction between generic and user code by registering new types and functions and creating new shell commands.

LENS is primarily a backpropagation simulator, designed for feed-forward, simple-recurrent, backprop-through-time, and fully recurrent networks. However, the basic network framework can be easily adapted to other models, and deterministic Boltzmann machines and Kohonen networks have been implemented as well. LENS operates on most Unix platforms and has recently been ported to Microsoft Windows. It was written in C and uses the Tcl/Tk libraries to support graphics and a shell interface.

Section A.1 of this document compares the performance of LENS to several other popular backpropagation simulators. Section A.2 explains some of the optimizations that lead to its good performance. Section A.3 briefly describes facilities for training on multiple machines in parallel. Section A.4 explains some of the principles that ease customization of LENS and Section A.5 considers its user interface.

A.1 Performance benchmarks

LENS[1] has been benchmarked along with five other commonly used, non-commercial simulators: SNNS[2], UTS[3], PDP++[4], RCS[5], and TLEARN[6]. The simulators performed backpropagation training using momentum descent on a feed-forward network having two hidden layers. The input and output layers each contained four units. The two hidden layers were approximately equal in size and were adjusted to control the total number of links in the network, which ranged from 100 to 1 million. The training set consisted of 40 random patterns and, where possible, batch learning was used. Thus, 40 forward and backward passes were performed before each weight update. All simulations were run on an unloaded 450 MHz Pentium II.

A.1.1 Memory usage

With medium to large networks, simulators are memory-intensive applications, and memory use is a critical factor in their performance. Time spent on memory operations can dominate that spent on floating-point operations. Although the use of small structures allows larger networks to fit into main memory, a reasonable simulator will not push the limits of current machines even with a network having several million links. A more important result of conservative memory use is its contribution to cache performance. Simulator speed is primarily bounded by the rate at which the program can cycle through the links. Using small link representations leads to fewer cache misses and greater speed. Due to interactions between the network's working set size, the sizes of primary and secondary caches, and the cache replacement protocols, simulator speed is not a linear function of network size and is not easily predicted based on the machine's floating-point performance.

Figure A.1 shows the memory requirements of the six simulators. With small networks, the memory is almost entirely due to simulator overhead. Next to PDP++, which has a very large profile, LENS has the second highest at just under 1.7 MB. Nevertheless, the overhead of most programs does not affect their working set size, and memory usage is not a critical factor on small networks.

As the networks grow, the memory devoted to link structures begins to dominate. Here the important differences between the simulators become evident. LENS uses 3 (4-byte) words per link: one for its weight, one for its error derivative, and one for the previous weight change, which contributes the momentum term.[7] SNNS and RCS each appear to use 6 words, which is reasonable. PDP++, TLEARN, and UTS, on the other hand, use roughly 10, 34, and 35 words, respectively.

Although not a factor in these experiments, the amount of memory devoted to data sets can also be an important issue. Experiments on language can involve corpora consisting of hundreds of thousands or millions of examples. LENS is able to reduce the memory and time required for example sets by using a mixture of dense and sparse representations, loading examples on-the-fly from a file or pipeline, and drawing examples from biased distributions.

[1] LENS, v. 2.02, was run in batch mode. It is available at http://www.cs.cmu.edu/~/Lens.

[2] SNNS, v. 4.2, is the Stuttgarter Neural Network Simulator from the University of Tuebingen, Germany. It was run with the batchman program using the BackpropMomentum learning function. It is available at http://www-ra.informatik.uni-tuebingen.de/SNNS.

[3] UTS, v. 4.1p1, is the sequel to XERION, and was developed at the University of Toronto. It is available at ftp://ftp.cs.toronto.edu/pub/xerion.

[4] PDP++, v. 1.2, was developed at Carnegie Mellon University. bp++ was run in -nogui mode. It is available at http://www.cnbc.cmu.edu/PDP++/PDP++.html.

[5] RCS, v. 4.2, is the Rochester Connectionist Simulator, developed at the University of Rochester. It was run using 6 settling steps and online learning. It is available at ftp://ftp.cs.rochester.edu/pub/packages/simulator.

[6] TLEARN, v. 1.0, was developed at the University of California, San Diego. It is available at http://crl.ucsd.edu/innate/tlearn.html.

[7] LENS can also be compiled with flags that create a fourth field for each link, which allows the delta-bar-delta algorithm to be used.

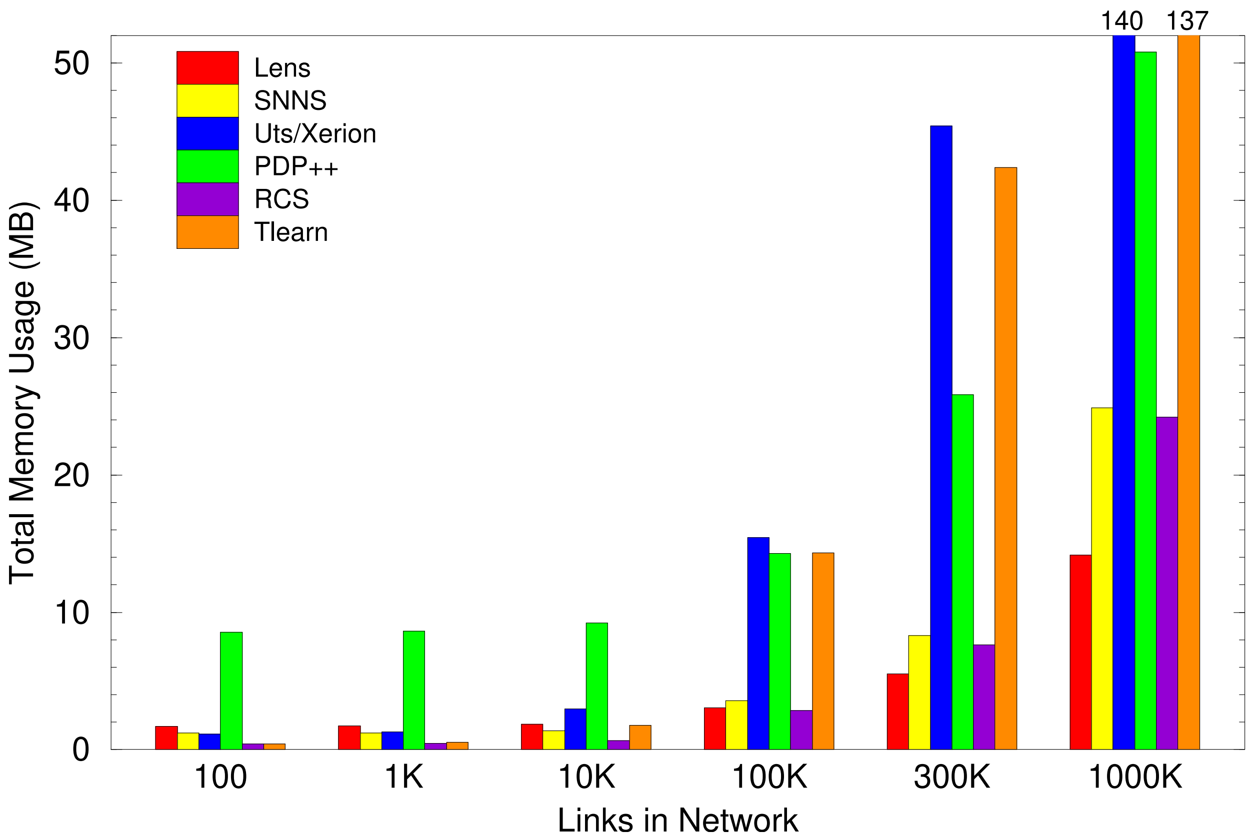


Figure A.1: Total memory usage with networks of varying size.

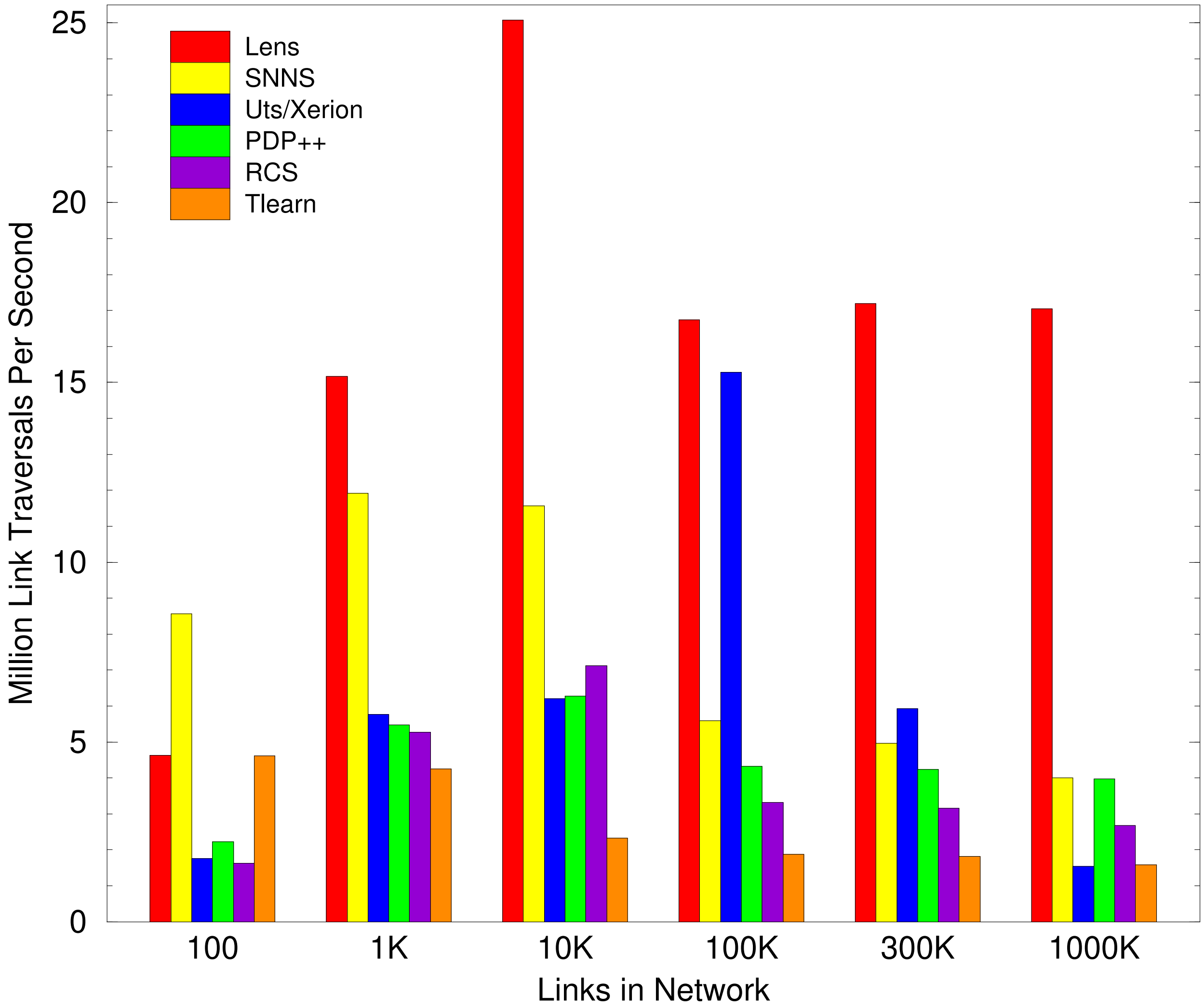


Figure A.2: Speed as measured in millions of link traversals per second during training of a feed-forward network.

A.1.2 Speed

Simulator speed was evaluated by training the networks for a number of weight updates equal to 10 million divided by the number of links in the network. A *link traversal* occurred for each link once in the forward pass, once in the backward pass, and once during a weight update. Thus, all networks were trained for 810 million link traversals. Training time included loading and shutting down the simulators. Three trials were run for each simulator on each size of network and the best of the three times was used. The results are shown in Figure A.2.

On the smallest network, LENS performs only moderately well. Tlearn does as well and SNNS is nearly twice as fast due to smaller overhead outside of the inner link-traversal loop. However, LENS was not optimized for networks of this size. On larger networks, which are increasingly used in connectionist modeling, it is the clear winner. At 300,000 links LENS is over three times faster than the competitors and at a million links it is over four times faster. This efficiency with large networks is particularly important for the CSCP model, which uses 2.9 million links.

Each simulator tends to have a sweet spot at which the relatively lower overhead of a large network and the better cache performance of a small network result in optimal link traversal rate. On this machine, LENS appears to have a peak at around 10,000 weights, while UTS has a surprisingly strong sweet spot at 100,000 links. These results are replicable on similar machines but may shift significantly on other architectures.

A.2 Optimizations

This section discusses some of the design principles and optimizations that result in the good performance of LENS on medium to large networks. To begin with, units in a network are organized into groups. The groups tend to correspond directly to layers of the network. All units in a group are of the same type, using the same input and output functions. In most models, units calculate their input as the dot-product of the incoming weight vector and the vector of corresponding unit activations and their output as the sigmoid of their input. In the backward pass the reverse processes occur. The unit calculates its input derivative (the derivative of the error w.r.t. the unit's input) as the product of its output derivative and the derivative of the input w.r.t. the output. It then sends error derivatives back to its incoming links and to the output derivatives of the sending units. Computing the unit inputs and sending back the input derivatives comprise the inner loops of backpropagation training, and they will typically use around 99% of the processing time on a large network.

In optimizing the inner loops, it is important to minimize the number of memory accesses they require by properly structuring the link representations. There are two basic ways to organize the links: They could be controlled by their sending unit (the one from which the link projects) and activation "pushed" over the link, or links could be controlled by the receiving unit and the activation "pulled" over the link. LENS uses the latter method. Each unit maintains the arrays of its incoming links but has no direct access to its outgoing links. During the forward pass, as a unit traverses its incoming links it need only access the link weight and the sending-unit activation from memory. The accumulating input value can remain in a register. Thus, there are just two memory reads per link. If links were owned by the sender and the values were pushed to the receivers, the output of the sending unit could remain in register but the input to the receiving unit would have to be retrieved from memory, incremented, and then stored back, requiring an extra memory access.

Because the values propagated on the backward pass depend on the receiving unit's input-combining function, which need not be a dot-product, it is easier for the receiving unit to push the derivatives backward over its incoming links. This involves six memory accesses: the link weight and the sending-unit output are loaded and the link derivative and sending-unit output derivative are incremented. If one knew that all receiving units used the dot-product, this could be reduced to 5 accesses by having the sending unit pull values backward across its outgoing links, but this would likely result in worse cache performance.

Although a receiving unit could get projections from several different groups or receive sparse inputs, projections tend to arise from a few blocks of consecutive units. LENS uses this block structure to its advantage. Because we know, by definition, that all sending units in a block are consecutive and belong to the same group, if those units are allocated in a single array of memory we could simply walk down the array as we access the output or output derivative of each unit.

The four critical values that must be retrieved in the backward pass are the links' weights and derivatives and the sending units' outputs and output derivatives. To get good performance, we would like to be able to fill the cache with as many of these values as possible. Therefore, the weights and derivatives of incoming links to each unit are stored in a single array. The last weight change and any other less critical values for the link are stored in a separate array. Rather than accessing the output and output derivative directly from the sending-unit structures, which would involve loading the entire sending-unit array into the cache, we keep separate arrays for each group that just contain a copy of the units' outputs or output derivatives. This minimizes the amount of non-critical information that enters the cache.

To further improve the speed, the inner loops are unrolled ten times by hand, which is noticeably better than any unrolling that may be performed by most optimizing compilers.

Finally, although unit-level costs are relatively insignificant for large networks, some improvement can be gained by using a fast sigmoid function. In place of the straight-forward sigmoid, which requires an exponentiation and a division, LENS uses a lookup table of 32K values and performs linear interpolation between values. This is accurate to within 2×10^{-7}. Although the fast sigmoid can speed up small networks by 15-20%, its effect on large networks is minimal.

A.3 Parallel training

LENS provides utilities for training networks in parallel on multiple machines. Parallel training is at the batch-level. That is, the network itself is not partitioned among machines. Each machine has a complete copy of the network and its own example set. In order for parallel training to be effective, the batch size must be large enough that the work can be partitioned among the clients without the overhead of communication dominating any benefits of parallelism. Nevertheless, some networks do see a performance advantage. There are two different forms of parallel training: *synchronous* and *asynchronous*.

Synchronous training is functionally equivalent to single-processor batch learning. At the start of each batch, the server sends a copy of the network's link weights to each of the clients and tells each client how many examples to process. The clients run the network on the assigned examples, accumulating link derivatives, and then ship the link derivatives back to the server. When the server has summed the derivatives from each of the clients, the weights are updated and the process repeats. A potential drawback of synchronous training is that it is only as fast as the slowest client. However, LENS maintains a running estimate of the speed of each client and adjusts the assignments so that all machines complete in approximately the same amount of time, assuming their loads are stable. A more serious drawback is that the clients are idle for the time it takes the server to update the weights and either send or receive from the other clients.

In the second form, asynchronous training, each client is given the same size batch of examples to run. When a client is done, it returns its derivatives and the server immediately updates the weights and sends back the new weight information. The primary advantage is that clients need not wait for one another unless the server is really overloaded. However, the drawback is that the clients are working with slightly different versions of the network and training can be unstable at high learning rates or with large batch sizes.

Although parallel training is useful when a single network must be trained as quickly as possible and machines are available, it is less useful than one might expect. It is typically the case when working with neural networks that a range of network or training parameters must be searched to find the best performance. Therefore, users rarely want to run just one network. In this case, it is more efficient to simply devote each machine to its own network, thus obtaining perfect parallelism. This was the approach used in training the three instances of the CSCP model.

A.4 Customization

The novel way in which LENS organizes unit-level functions provides considerable flexibility without the need to modify the program. Operations on units are divided into three main classes of procedures: for computing the unit input, computing the unit output, and attributing cost or error directly to the unit. For each of these operations, the type of the unit's group defines a pipeline of simple procedures. The procedures can be combined to produce various behaviors.

For example, most units will have a basic output procedure, such as linear, sigmoidal, or exponential, which determines the output as a function of the input. Once that is computed, a secondary procedure in the output pipeline might inject noise. Another procedure could then integrate the unit's output over time or normalize the outputs across a group. Each procedure has a corresponding inverse procedure which operates in the backward pass to compute the unit's input derivative from its output derivative. Without the ability to combine simple operations in this way, one would have to create many more group types to handle all reasonable combinations of simple procedures.

However, no simulator can satisfy all users so that many modelers will eventually need to get inside and make their own changes. In most simulators, changes would need to be made in various places throughout the code, which leads to problems whenever a new version is released. LENS was largely born out of frustration with the difficulty of tracing and modifying the code in other simulators. To ease customization, LENS provides an *extension* module, in which user modifications can be contained. Three main features of its design facilitate encapsulation of changes: extension structures are provided to allow the user to augment the major structures, network behavior is controlled by a modifiable hierarchy of function pointers, and new function types for controlling various aspects of the simulator can be registered to make them easily accessible from the command interface.

The network maintains pointers to functions for such actions as training for a number of weight updates, training on

```
#define SINE_OUT ((mask) 1 << 20)

static void sineOutput(Group G, GroupProc P) {
  FOR_EACH_UNIT(G, U->output = sin(U->input););
}
static void sineOutputBack(Group G, GroupProc P) {
  FOR_EACH_UNIT(G, U->inputDeriv = U->outputDeriv * cos(U->input));
}
static void sineOutputInit(Group G, GroupProc P) {
  P->forwardProc  = sineOutput;
  P->backwardProc = sineOutputBack;
}
flag userInit(void) {
  registerGroupType("SINE_OUT", SINE_OUT, GROUP_OUTPUT, sineOutputInit);
  return TCL_OK;
}
```

Table A.1: The code to add a custom unit output function.

a single batch of examples, training on an example, training on an event within an example, and so forth. The functions tend to become simpler as we descend the hierarchy and each function typically contains a loop that repeatedly makes calls using a pointer to the function below it. Many changes to network behavior can be made by replacing a single network function, minimizing the amount of new code that must be written and enabling the new code to remain in the extension module. Having written a new function, the user might then create a shell command that causes the network to use the new function in place of the old one.

However, to make new network or group types more naturally accessible from the shell interface, types can be *registered*. This creates a name by which the user will be able to refer to that type, making it equivalent to the built-in types. Rather than creating a special shell command, a new network type could be created and an initialization procedure defined to configure any new networks of that type. Customizations that currently may be registered include additions of basic network types, unit input, output, and cost functions, and algorithms for updating the weights, for selecting the next example, and for creating link projection patterns. Table A.1 contains all of the code necessary to create and register a new unit output function that computes the sine of its input, assuming that might be useful for some reason.

A.5 Interface

The primary interface to LENS is a Tcl/Tk-based command language. The user can either enter commands to a shell or run programs out of script files. Over 120 commands are currently available allowing the user to, among other things, build and lesion networks, save and load example and weight files, describe the network layout, and, of course, train and test. New procedures can be written in Tcl, which feature is especially helpful in running experiments. Commands can also be written in C and compiled if speed is an issue. Finally, the fields in the C structures of the network and example sets can be accessed from the shell.

The design of LENS was guided by the philosophy that common things should be easy and difficult things should be possible. One useful feature is the relative ease with which networks can be constructed. Most feed-forward and simple-recurrent networks can be described with a single command. For example, the command:

```
addNet myNet 10 20 ELMAN 5 SOFT_MAX
```

would create a simple-recurrent network with 10 input units, a 20-unit hidden layer with corresponding context layer, and a 5-unit output layer which uses a soft-max constraint and the appropriate divergence error measure. The input and context groups project to the hidden layer which projects to the output layer. To build the same network, other simulators might require that six or more commands be issued, a C program be written and compiled, or that the network be constructed by pointing and clicking on a graphical interface. If more complicated networks are desired in LENS, such as ones that use non-simple recurrence or sparse connectivity, the network can be partially built with `addNet` and then extended piece-by-piece.

The script language makes it possible to parameterize aspects of network building. For example, a procedure might be defined to create a network with a specified number of hidden units, thus making it easy to experiment with different architectures. Some simulators require that a separate file be created for each architecture, which can be a serious hindrance.

A.5.1 Displays

Although LENS can be operated using only the shell, graphical interfaces are convenient for providing better visualization and quick access to common operations. By default, a main window, which gives access to the most useful commands and training parameters, is opened. The eight panels in the main window can be individually hidden to conserve screen space. A shell console window is also optional and provides a nicer command-line environment than the basic Tcl shell, including the ability to edit commands, traverse the command history, perform command- and file-name completion, and execute commands while LENS is busy. The *object viewer* allows the user to view and edit the C data structures which represent the networks and example sets and their components. Where appropriate, fields in the C structures can be hidden or write-protected.

The *unit viewer* is perhaps the most useful display. It shows the training or testing examples and the activations, inputs, derivatives, or other values associated with the units or the links projecting to or from a single unit. This is helpful in observing the behavior of the network and quickly diagnosing problems. By default, the layout of the network in this window will be created automatically, but commands are also provided for customizing the network representation. The *link viewer* depicts the values associated with some or all of the links and calculates summary statistics. Finally, any real-valued field, typically the network's error, may be graphed over time.

A.6 Conclusion

LENS is a fast, flexible neural network simulator with the potential to satisfy the needs of a wide variety of users. Although currently used mainly by experienced modelers, the relatively straightforward interface, the ease of creating new simulations, and the ability to run on a variety of platforms make it well suited for use in introductory courses. LENS is available free-of-charge to those teaching or conducting research at academic institutions. The complete manual and installation instructions can be found on the web at...

http://www.cs.cmu.edu/~dr/Lens

Appendix B

SLG: The Simple Language Generator

This appendix introduces the design and use of the Simple Language Generator (SLG), which is responsible for producing and parsing sentences in the Penglish language. SLG enables the user to construct small but interesting stochastic context-free languages with relative ease. Although context-free grammars are convenient for representing natural language syntax, they do not easily support the semantic and pragmatic constraints that make certain combinations of words or structures more likely than others. Context-free grammars for languages involving many interacting constraints can become extremely complex and cannot reasonably be written by hand. SLG allows the basic syntax of a grammar to be specified in context-free form and constraints to be applied atop this framework in a relatively natural fashion. The program then converts this combination of grammar and constraints into a standard stochastic context-free grammar for use in generating sentences, in parsing, or in making context-dependent likelihood predictions of the sequence of words in a sentence.

Let us first take a step back and explain why such a program might be interesting and useful. A common goal in the field of machine learning is the development of models that are able to capture the structure of a language, be it a natural, human language or something more abstract. This might serve as a foundation for a system which learns the rules of grammaticality based on example or one that is designed to comprehend and produce messages in the language. Although it is often desirable to work directly with a complete natural language, in studying the behavior of a particular learning method or in comparing multiple strategies it is sometimes necessary to have at our disposal languages with well-understood and easily controlled properties.

To produce such a *pseudo-language*, we typically rely on a grammar which defines the legal strings, or sentences, it contains. A *generative* grammar is one that can produce those sentences. With most reasonable languages, it is usually not very difficult to write a program to generate syntactically legal strings in the language. However, language generation is a much harder problem if one is concerned with violations of semantic or pragmatic acceptability. One possible method might be to generate syntactically valid sentences and then to filter out those that violate a set of semantic constraints. But such a technique would not work well if we are concerned with producing sentences whose frequencies obey some desired probability distribution.

An aim of many researchers is to train a neural network, hidden markov model, or other learning method to model a language. This generally translates into the ability to perform accurate word prediction at any point in a sentence. In order to train such a model on a pseudo-language, we will need to be able to generate a language whose statistics obey hard and probabilistic constraints from both syntax and semantics. And in order to evaluate such a model, we will need to be able to produce the theoretically correct word predictions, given the grammar, to compare with the model's prediction. Although there are many ways to generate a language, most of them do not enable the designer to control both syntactic and semantic constraints on probability and even fewer permit the rapid calculation of word-by-word predictions given the grammar.

A simple yet fairly powerful form of grammar, and one that seems quite suited to representing the structure of most natural languages, is the context-free grammar, or CFG (see Hopcroft & Ullman, 1979, for an introduction). By specifying probabilities that each possible *production*, or transition, in the grammar is used, we can control the distribution of sentences produced by a CFG. Thus we form a type of generative grammar known as a stochastic context-free grammar, or SCFG. The advantage of the SCFG is that it has been the subject of considerable analysis, and we have reasonably fast algorithms for parsing and producing word predictions using SCFGs.

Unfortunately, if one is interested in designing complex languages by hand, the SCFG can be rather cumbersome. In order to produce a language of significant complexity, where *complexity* is used in a non-technical sense, the required grammar becomes long and complicated, involving considerable redundancy and a host of symbols, which often have rather abstract relationships to the final language. In the SCFG, probabilities must be specified for each production in the grammar, but reasonable probabilities are difficult to determine by hand if the symbols involved do not have clear mappings to well-understood properties of the intended language. Therefore, designing interesting SCFG grammars by hand becomes quite impossible.

The goal of SLG, or the Simple Language Generator, is to allow a human to specify relatively concise and intuitive grammars which nevertheless define interesting languages. The grammar interpreted by SLG is similar in basic form to an SCFG, but it allows the designer to specify additional constraints that alter the resulting language. The ability to reuse constraints helps to eliminate redundancy. SLG can then convert the user's grammar to a standard SCFG in a process known as *resolving* the grammar. Once we have obtained an SCFG that is equivalent to the original grammar, albeit much longer and more complex, we can easily generate sentences in the language or produce optimal word predictions.

This report explains the use of SLG and some of its inner workings. Section B.1 describes the grammar specification language. Section B.2 explains the process by which constraints are resolved. Section B.3 explains the process of reducing or minimizing the size of grammars. Section B.4 explains how parsing is performed by SLG and the conversion of a grammar to Chomsky normal form which is required for this. Section B.5 describes the method of converting grammars to Greibach normal form and how this is used to produce word predictions. Finally, Section B.6 mentions some possible future extensions to the program and provides the address for downloading SLG.

B.1 The grammar

The grammar interpreted by SLG[1] is a superset of a standard SCFG grammar. Therefore, any ordinary SCFG, and hence any finite-state machine, can be handled in a straightforward manner. One can view the process of generating a sentence with an SCFG as the branching of an inverted tree. Each non-terminal symbol branches into the symbols in its chosen production. The grammar is context-free because the branching of each symbol depends only on the symbol itself and is unaffected by context, or the symbols around it.

While CFGs are a convenient way to capture the syntax of many languages, and have thus attracted the attention of linguists, if we are concerned with the frequency of sentences, we must consider the semantics and pragmatics of natural languages, which play an important role in the choice of productions. Unfortunately, semantics and pragmatics essentially introduce context-sensitive constraints into the grammar. Producing the contents of an NP that serves as a direct object may depend on the verb as well as its subject and other modifiers. The question is how can we introduce this type of information into an SCFG without completely restructuring it, which would destroy the nice, simple model of syntax that it provides.

What we would like to be able to do is to write an SCFG which describes the syntax of the language but then to constrain the behavior of one symbol given the productions of one, or more, other symbols in the tree. For example, when producing natural sentences, we might want to constrain the choice of subject based upon the choice of verb or the choice of adjective based on the noun it is modifying. Given an SCFG and a list of such constraints, it is possible, although not entirely straightforward, to generate sentences that satisfy both the grammar and the constraints (as long as they are not circular) without restructuring the grammar.

However, given a grammar like this in which the structure and constraints are separate, it would be next to impossible to efficiently parse and generate word predictions. To do that, we would really like just a standard SCFG. The idea behind SLG is that we can take a grammar consisting of an SCFG base with additional constraints and convert it into a new SCFG with all of the constraints embedded in the context-free productions. It does this by determining which portion of the CFG tree is affected by each constraint and restructuring just those portions of the CFG. This process is explained in Section B.2. We begin here by simply explaining how an SLG grammar is written and how the constraints are specified.

[1] This report is based on SLG version 2.0.

Defining an SLG constraint involves two steps. First, a constraint function is defined which specifies how the choice of productions from the constraining symbol, or the *source*, affects the choice of productions of the constrained symbol, or the *goal*. Then the constraint must be applied to each appropriate pair of source and goal. This is done by specifying which subtree or subtrees are affected by the constraint. The subtree begins at the *root*, or the symbol which is the lowest mutual-ancestor of the source and goal. Typically, noun-phrases and verb-phrases appear in many places in a natural-language grammar. The ability to reuse a single subject-verb constraint function for each pair helps eliminate redundancy and segregate semantic/pragmatic information from syntactic information in the grammar. Because resolving a constraint only involves altering the paths through the tree that start at the root and extend to the source or goal, the use of constraints does not render the SLG grammar super-context-free in the theoretical sense.

B.1.1 Using the SLG grammar

Figure B.1 contains a sample SLG grammar, illustrating its syntax and some of the available features. The first 11 lines contain symbol definitions. A symbol definition begins with a list of the symbols to be defined, separated by | characters. It is convenient to read the | character as "or". A symbol name can consist of any string of characters excluding white space and the following special characters: `; | : , { } ( ) !` Alternately, a symbol name can be any string enclosed in double quotes. This allows multi-word symbols and symbols using the special characters. The first symbol defined becomes the start symbol.

```
1    S  : CNP VP "." |
2      {LegalIntVerb, CNP NP.* N, VP VI} |
3      {LegalTrnVerb, CNP NP.* N, VP VT};
4    VP : VI | VT CNP (0.7) |
5      {LegalObject,  VT, CNP NP.* N};
6    NP | NP2 : the N;
7    CNP : NP | NP and NP2 |
8      {DontRepeatNoun, NP N, NP2 N};
9    N : boy (0.3) | cat (0.3) | dog;
10   VI : barked | slept;
11   VT : bit | fed;
12
13   LegalIntVerb {
14     boy | cat : slept;
15     dog       : barked (0.8) | slept;
16   }
17
18   LegalTrnVerb {
19     dog | cat ! fed;
20   }
21
22   LegalObject {
23     bit | fed : boy (0.6) | cat (0.2) | dog;
24     fed       ! boy;
25   }
26
27   DontRepeatNoun {
28     boy ! boy;
29     cat ! cat;
30     dog ! dog;
31   }
```

Figure B.1: A grammar for producing simple sentences.

Each of the symbols in the definition list will be synonymous. That is, they will have the same set of productions and will be the *root* symbol for the same sets of constraints. It may not seem particularly useful to have equivalent symbols, but it often comes in handy when one needs to apply different constraints to otherwise identical symbols, and it can sometimes help make the grammar more clear. An example of a shared definition is that of "`NP | NP2`" on line 6 of the grammar. Because we created two different non-terminals for producing nouns, we can distinguish between them in the constraint on line 8.

Following the definition list is a colon and then a list of *productions* and *constraints*, separated by |'s. The definition is terminated with a semicolon. A production is a string of symbols (separated by white space) followed by an optional probability (P) enclosed in parentheses. The probabilities for all possible productions from a symbol must sum to 1.0. Any productions whose P is not specified will be given the same P, which is calculated such that the

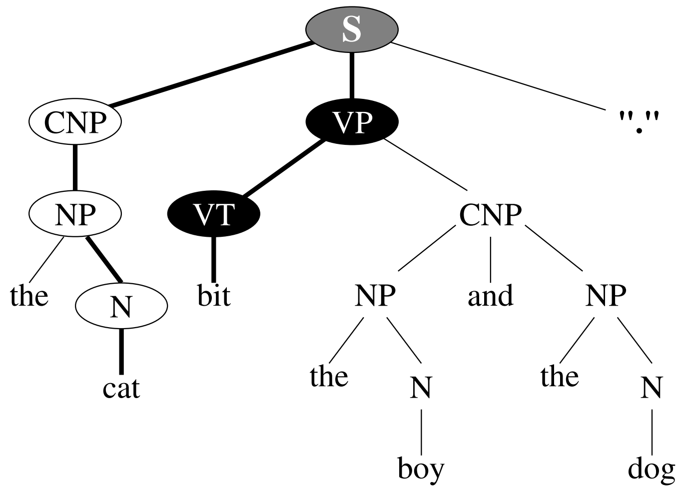


Figure B.2: Parse tree from the grammar with a source path shown in white and a goal path shown in black.

overall sum becomes 1.0. Therefore, if three productions are defined and P = 0.6 is specified for the first and no P is specified for the others, the other productions will default to 0.2.

Constraints, such as {`LegalTrnVerb, CNP NP.* N, VP VT`}, are enclosed in curly braces and consist of three parts, separated by commas. The first specifies the constraint function, which must be defined separately. The second specifies the *source path* and the third specifies the *goal path*. When a sentence is generated with a CFG, we can view the process as the branching of a tree, beginning with the start symbol, which is `S` in this case. Figure B.2 illustrates the parse tree for a sentence generated by the example grammar.

When a constraint is given in a symbol definition, the symbol currently being defined is called the *root* of the constraint. For example, the root of the constraint on line 8 is `CNP`. Note that the root of a constraint need not be the start symbol of the grammar. A constraint path consists of a series of symbols or regular expressions separated by white space. It matches a path through the states in the parse tree which begins at the root symbol. Each of the symbols or regular expressions in the path must match a symbol in the parse tree at the next level down. If either the source path or the goal path does not match a path in the tree, the constraint does not apply.

In the case of the tree in Figure B.2, the constraint {`LegalTrnVerb, CNP NP.* N, VP VT`}, which has root `S`, is applicable. The source path is marked with white ovals and the goal path with black ovals. The last symbol in the source path is called the *source*, because it will be the source of the constraint. In this case, the source is `N`. The last symbol in the goal path, `VT`, is called the *goal*. A constraint is only valid if the first symbol on the source path is different from the first symbol on the goal path and the two symbols appear together in at least one root production. In this case, both `CNP` and `VP` appear in the `S` production `CNP VP "."`. Additionally, the root symbol itself may not appear on the source or goal paths except as the first symbol. Note that `NP.*` is a regular expression that matches any symbol whose name starts with `NP`.

The meaning of the constraint is that the choice of production out of the source, or the production that the source symbol performs, will constrain the choice of production out of the goal. It does this using the specified constraint function, which must be defined separately. A function definition consists of the name of the function followed by a list of terms enclosed in curly braces. Each term begins with a list of productions called the *source list*. These must be valid productions out of the source, as given by the definition of the source symbol. Note that the elements in this list are *productions*, not just symbol names. In the current example, all of these productions happen to consist of a single symbol, but in general it is possible for constraints to involve more complex productions comprising a series of symbols. Following the source list is either a colon or exclamation point and then the *goal list*, which is a list of possible productions out of the goal symbol, given that the source symbol uses a production in the source list.

If the character separating the source and goal lists is an exclamation point, rather than a colon, then the goal list specifies productions which *cannot* be taken from the goal symbol if the source symbol produces one of the productions in the source list. This is a convenient way to eliminate selected productions. For example, the `LegalTrnVerb` function says that if the subject is dog or cat then the transitive verb cannot be fed. Probabilities should not be specified when using an exclamation point.

On the other hand, if the character separating the source and goal lists is a colon, the goal list specifies the only legal productions from the goal symbol if the constraint term applies. In addition, probabilities can be specified for the goal list productions that will modify the distribution of productions produced by the goal. The new distribution does not replace the old one. Rather, it *filters* the distribution. When one distribution filters another, the corresponding terms are multiplied and the results re-normalized. For example, symbol N normally produces boy, cat, or dog with respective probabilities of 0.3, 0.3, and 0.4. When filtered by the first term in the LegalObject function, which specifies a distribution of 0.6, 0.2, 0.2, the resulting distribution becomes 0.5625, 0.1875, 0.25. Note that if a production has P = 0.0 in either distribution, it will have P = 0.0 in the result. If no probabilities are specified in the goal list, the constraint will eliminate goal productions that are not listed but will not change the relative likelihood of the listed productions.

Although it is usual for each source production to appear in at most one term in a constraint function, it is sometimes useful to define multiple terms for a production. For example, you might have one term that represents animals and one that represents fierce things, both of which include dog. If a source production matches more than one term then the goal productions are filtered by each of the matching terms. The order in which filtering occurs does not matter.

Let us examine the constraints employed in the example grammar. LegalIntVerb is used to constrain the choices of intransitive verbs given the subject or subjects of the sentence. Note that the constraining (source) path in this constraint, CNP NP.* N, will always be matched because an S must produce a CNP, which must produce an NP (and possibly an NP2), which in turn must produce an N. The constrained (goal) path, VP VI, may not be matched because a verb phrase may contain a transitive verb rather than an intransitive one. If the constrained path does match, then this constraint will affect the choices of intransitive verb based on the choice of noun for the subject.

The definition of the LegalIntVerb function specifies that if the subject is boy or cat then the intransitive verb must be slept. However, if the subject is dog, then the intransitive verb will be barked with P = 0.8 and slept with P = 0.2. The LegalTrnVerb function is used to constrain the possible transitive verbs given the subject or subjects. If the subject is either dog or cat, the transitive verb cannot be fed. Therefore, it must be bit. Note that boy is not listed in any of the terms in the definition of LegalTrnVerb. Therefore, if the subject is boy, the constraint has no effect and the transitive verb will be bit or fed with equal likelihood. If the subject is something like, "the boy and the dog", then both will constrain the verb. Since the boy does not affect the verb, only the constraint for the dog will apply.

The LegalObject function is used to constrain the choices of object given the transitive verb. The first term says that if the verb is bit or fed, all nouns are possible, but their probabilities have been altered. The second term, however, specifies that if the verb is fed, the object cannot be boy. In this case the P of the object becoming cat will be $3/7$ and the P of dog will be $4/7$. Finally, the DontRepeatNoun constraint eliminates the possibility of a compound noun consisting of the same noun twice, such as "...the dog and the dog".

The amount or type of white space does not matter in the SLGgrammar file, except that any line starting with a # will be treated as a comment.

B.1.2 Other features

One useful feature that has not been mentioned is the *epsilon* production. A standard concept in CFGs, these are productions that generate nothing and thus eliminate the current symbol from the tree. While these do not alter the theoretical power of the grammar, their use can simplify grammars. Consider the two examples shown in Figure B.3. These are equivalent grammars that produce a noun-phrase containing an optional article and optional adjective before the noun. The first grammar does not use epsilon productions and therefore must specify all four possible types of noun-phrase. The second grammar uses epsilon productions, which are written as an empty pair of double-quotes, to make the article and adjective optional while freeing the user from enumerating every possibility.

Finally, constraints may be given a priority which determines the order in which they will be resolved. By default, if there are no other dependencies, constraints are handled in some arbitrary order when resolving the grammar. However, sometimes the resolution process goes faster if certain constraints are resolved before others. The user can influence the order of resolution by giving constraints a priority. By default, constraints have priority 0, but the priority

```
NP : N | ART N | ADJ N | ART ADJ N;
ART: the | a;
ADJ: green | putrid;
```

```
NP : ART ADJ N;
ART: "" | the | a;
ADJ: "" | green | putrid;
```

Figure B.3: Using epsilon productions to simplify a grammar.

can be changed by placing a fourth field in the constraint specification, as in {`foo, NP N, VP VT, 3`}. Higher priority constraints will be resolved first. The priority may be negative. For most grammars, specifying priorities will have no noticeable effect. It is most useful if you would like to observe the intermediate stages of the resolution process under a particular constraint ordering.

B.1.3 Limited cross-dependency

A main attraction to CFGs has been their ability to conveniently capture center-embedding, which is a common feature of English and most other languages. A nested center-embedded sentence might have the general structure `N1 N2 N3 V3 V2 V1`, where `V1` depends on `N1`, `V2` on `N2`, and so on. These are easily captured by CFGs. However, of considerable interest and trouble to the linguistic community has been the existence of a few languages, most notably Dutch and Swiss German, that permit cross-dependencies (Christiansen & Chater, 1999b), which have the structure `N1 N2 N3 V1 V2 V3`. These cannot in general be described by a CFG and, even if the depth of the embedding is limited, are difficult to describe once agreement and semantic constraints are introduced.

```
S: N1 N2 N3 V1 V2 V3 |
   {N-V, N1, V1} | {N-V, N2, V2} | {N-V, N3, V3};
N1 : dog | dogs | cat | cats;
N2 | N3 : dog | dogs | cat | cats | "" (0.8);
V1 | V2 | V3 : barks | bark | purrs | purr | "";

N-V {dog:barks; dogs:bark; cat:purrs; cats:purr; "":"";}
```

```
S:  N NP V VP | {N-V, N, V} |
    {NP-VP, NP, VP} | {N-V, NP N, VP V} |
    {NP-VP, NP NP, VP VP} | {N-V, NP NP N, VP VP V};
N:  dog | dogs | cat | cats;
V:  barks | bark | purrs | purr;
NP: N NP | "" (0.8);
VP: V VP | "";

N-V {dog:barks; dogs:bark; cat:purrs; cats:purr;}
NP-VP {"":""; N NP:V VP;}
```

Figure B.4: Two SLG grammars for limited cross-dependency.

However, cross-dependencies of limited depth are not too difficult to describe using an SLG grammar. Figure B.4 shows two ways in which, in rather simplified form, one might write such a grammar. The first uses a flat representation to explicitly allow up to three possible pairs. Three constraints are required to implement agreement. The use of the epsilon production allows `N2` and `N3` to be optional, and the constraints prevent the corresponding verbs from appearing in the absence of the nouns. When resolved into an SCFG, this grammar requires 25 non-terminal symbols and 124 productions.

The second example uses a nested structure, which may be more convenient and more linguistically reasonable in a full grammar. In this case, two constraints are used for each possible depth of embedding. However, the grammar is not entirely adequate since it could produce embeddings beyond depth two which would not be subject to the constraints. Because it is formally equivalent to an SCFG, it is not possible to implement unbounded cross-dependency in an SLG grammar.

B.1.4 A larger example

Figure B.5 shows a much more complex SLG grammar which produces some reasonably interesting English sentences. Once it is resolved, this grammar produces an SCFG with 140 non-terminals and 442 productions, which is considerably larger than the original grammar. Some sentences produced with this grammar are listed in Figure B.6.

B.2 Resolving the grammar

This section explains the process by which SLG takes a grammar involving constraints and transforms it into a grammar in standard SCFG form. It does this by *resolving* each of the constraints. Resolving a constraint is a rather complex process, but essentially involves splitting each of the symbols along the source and goal paths into sub-symbols, which correspond to terms or conjunctions of terms in the constraint function. The tricky part comes in correctly computing the production probabilities that go along with all of this.

B.2.1 Resolving a simple constraint

Consider the grammar depicted in Figure B.7. The source symbol produces either `A B` or `A C`, `A` produces `i`, `j`, or `k`, and `B` produces `x` or `y`. However, when `S` produces `A B`, the production out of `A` should constrain the production out of `B`. The constraint function, `foo`, indicates that when `A` produces an `i`, `B` will produce an `x` with probability (P) 0.27273 and a `y` with P = 0.72727 (after filtering the base `B` distribution). But if `A` produces a `j` or `k`, `B` must produce `x`.

To resolve the constraint, we start at the source symbol, `A`. A sub-symbol is created for each term or set of terms that could be matched by a source production. In this case, no productions can match more than one term, but production `i` matches term 1 and productions `j` and `k` match term 2. Therefore, two new symbols are created. `A-1S0.1` only produces `i`, thus matching term 1, and `A-1S0.2` produces `j` or `k`, thus matching term 2. The relative frequency of `j` and `k` should not change, so `A-1S0.1` will produce `j` with P = 0.6 and `k` with P = 0.4. The *source strength* of each of the sub-symbols is the P that the original symbol, `A`, would have produced one of the productions in the sub-symbol. For example, the source strength of `A-1S0.1` is the P that `A` produces `i`, or 0.5. The source strength of `A-1S0.2` will be $0.2 + 0.3$, or 0.5 as well.

Now we take a step up toward the root. In this simple case, the source path was only one symbol long so we are now at the root. Each root production that matches the constraint will be split. A production matches this constraint if it contains at least one `A` and one `B`, so only the first constraint matches. Production `A B` will be replaced with a pair of productions, `A-1S0.1 B-1G0.1` and `A-1S0.2 B-1G0.2`. The P that `A-1S0.1 B-1G0.1` is used is equal to the product of the original P of `A B`, 0.5, and the source strength of sub-symbol `A-1S0.1`, also 0.5. This is the P that the original grammar would have produced an `i`, which is 0.25. The P that `A-1S0.2 B.1G0.2` is produced is the product of 0.5 and the source strength of `A-1S0.2`, which is also 0.25. Note that the overall P of producing a `j`, 0.3, has not changed.

What remains is to define the new `B` sub-symbols, `B-1G0.1` and `B-1G0.2`. `B-1G0.1` is the version of `B` that should be produced when term 1 of the constraint is satisfied. Thus, the distribution for `B-1G0.1` is the base `B` distribution, $(0.6, 0.4)$, filtered by the first term of `foo`, $(0.2, 0.8)$, which is be called the *constraining distribution*. The resulting distribution is $(0.27273, 0.72727)$. `B-1G0.2` must satisfy the second term of `foo`, which has a constraining distribution of $(1.0, 0.0)$, and therefore produces only `x`. Because symbol `B` itself is no longer used, the grammar reduction procedure (Section B.3) eliminates it. Figure B.8 shows the resulting fully-resolved grammar. For such a simple example, the use of the constraint didn't actually help reduce the size of the grammar.

B.2.2 Resolving a deeper constraint

If we extend the source and goal paths and allow multiple references to a source path symbol in a single production, the process of resolving a constraint becomes more complex. Consider the grammar in Figure B.9. The source path and goal path of the constraint now have two symbols each. Additionally, `S` can produce a production with two `A`'s

```
S  : SP VI . (.25) | SP VT OP . |
  {sub-intr, SP NP N, VI} | {sub-trns, SP NP N, VT} |
  {trns-obj, VT, OP NP N} | {sub-obj, SP NP N, OP NP N} |
  {intrans-ref, VI, SP RC VI};

SP | OP : NP | NP RC (.3) |
  {sub-intr, NP N, RC VI} | {sub-trns, NP N, RC VT} |
  {trns-obj, RC VT2, NP N};

RC : who VI | who VT OP | who SP VT2 |
  {trns-obj, VT, OP NP N} | {sub-trns, SP NP N, VT2};

NP : ART ADJ N | {noun-art, N, ART} | {noun-adj, N, ADJ};

ART: "" | the | a;

ADJ: "" (0.6) | quick | happy | hungry | nasty | mangy | crazy | sleazy;

N  : boy | boys | girl | girls | Mary | John | cat | cats | dog | dogs;

VI : walks | walk | bites | bite | eats | eat | barks | bark;

VT | VT2 : chases | chase | feeds | feed | sees | see | walks | walk | bites |
  bite;

sub-intr {
  boy  | girl  | Mary  | John  : walks | eats;
  boys | girls : walk  | eat;
  cat  | dog   : walks | bites | eats  | barks;
  cats | dogs  : walk  | bite  | eat   | bark;
  cat  | cats  ! bark  | barks;
}

sub-trns {
  boy  | girl  | Mary   | John : chases | feeds | sees(.1) | walks;
  boys | girls : chase  | feed     | see(.1) | walk;
  cat  | dog   : chases | sees(.2) | bites;
  cats | dogs  : chase  | see(.2)  | bite;
}

trns-obj {
  walk | walks : cat | cats | dog | dogs;
  see  | sees  : cat | cats;
}

sub-obj {
  Mary ! Mary;
  John ! John;
}

intrans-ref {
  walks | walk ! walks | walk;
  bites | bite ! bites | bite;
  eats  | eat  ! eats  | eat;
  barks | bark ! barks | bark;
}

noun-adj {
  boy  | boys | girl | girls | Mary | John ! mangy;
  John | cat  | cats | dog   | dogs ! sleazy;
}

noun-art {
  Mary | John : "";
  boys | girls | cats | dogs ! a;
  boy  | girl  | cat  | dog  ! "";
}
```

Figure B.5: A more complex SLG grammar.

```
a dog bites the happy boys .
dogs who chase the hungry cat bite nasty girls who walk .
the crazy cat walks .
the hungry cats who walk chase the hungry dog who chases the crazy girls .
the mangy dogs who walk bite the quick dog .
a nasty cat who sees the mangy cats bites .
girls chase the sleazy girls .
hungry cats eat .
the nasty cats bite Mary .
the nasty cat who a crazy girl chases bites the crazy boys .
```

Figure B.6: Sentences generated by the complex grammar.

```
S: A B | A C | {foo, A, B};
A: i (0.5) | j (0.3) | k (0.2);
B: x (0.6) | y (0.4);

foo {
  i     : x (0.2) | y (0.8);
  j | k : x;
}
```

Figure B.7: An SLG grammar with one simple constraint.

and A can produce a production with two C's. Finally, the production j from C satisfies two constraints. Each of these factors makes resolving the constraint more complex.

As before, we begin at the source symbol, C. Three sub-symbols will be created. C−1S0 produces only k, which does not match any terms. C−1S0.1 produces i, which matches term 1. C−1S0.1.2 produces j, which matches both terms. All three sub-symbols have a source strength of $1/3$. Now we step up the source path to symbol A. We will have to create six sub-symbols for A to cover all possible combinations of terms that might be satisfied by its productions. The first sub-symbol, A−1S1 should satisfy no terms. Therefore, its productions will be C−1S0, C−1S0 C−1S0, and E. The source strength of symbol A−1S1 is the sum of the *production strengths* for the three productions. A production strength is the original P of the production multiplied by the source strengths of any sub-symbols in the production. Thus, the production strength of E is just $1/3$. The production strength of C−1S0 is $1/3 \times 1/3 = 1/9$ and the production strength of C−1S0 C−1S0 will be $1/27$. The total source strength of A−1S1 will therefore be $13/27$. This is the P that the symbol A would not have produced an i or a j. The P of each production in A−1S1 will be the production strength divided by A−1S1's source strength. In other words, the productions are normalized.

The second sub-symbol produced, A−1S1.1, should lead to productions that satisfy term 1 of the constraint function. Thus, it should always produce a C sub-symbol which itself produces exactly one i. The production E is therefore dropped because it does not contain a C. Production C becomes C−1S0.1 with production strength $1/9$. Production C C is split into two productions, C−1S0.1 C−1S0 and C−1S0 C−1S0.1, each with strength $1/27$. Thus, either A−1S1.1 produces i, i k, or k i. The overall source strength of A−1S1.1 is $4/27$. To take one more example, sub-symbol A−1S1.2x1.2 should satisfy term 1 in two ways and term 2 in one way. Therefore, it must either produce i j or j i. It will have two productions, C−1S0.1 C−1S0.1.2 and C−1S0.1.2 C−1S0.1, each with strength $1/27$.

Now we step up to the root level, S. For each root production that contains at least one A and a B, we will create a sub-production for each combination of terms that we could satisfy by replacing the A's with various A sub-symbols, plus one production in which B is replaced by a sub-symbol that does not reach the goal. Production E will remain unchanged, but production A B will be replaced by seven productions.

```
S: A-1S0.1 B-1G0.1 (0.25) | A-1S0.2 B-1G0.2 (0.25) | A C (0.5);
A: i (0.5) | j (0.3) | k (0.2);
A-1S0.1: i;
A-1S0.2: j (0.6) | k (0.4);
B-1G0.1: x (0.27273) | y (0.72727);
B-1G0.2: x;
```

Figure B.8: The simple grammar with the constraint resolved.

```
S: A B | A A B | E | {foo, A C, B D};
A: C | C C | E;
C: i | j | k;
B: D | E;
D: x | y;

foo {
  i | j : x (0.2) | y (0.8);
  j : x;}
```

Figure B.9: A different grammar with a moderately complex constraint.

The first of these is `A B−1N1`, where `B−1N1` is a newly created sub-symbol of `B` that does not complete the goal path and is therefore not subject to the constraint. In this case, the goal path is `B D` so `B−1N1` cannot produce a `D` and therefore produces just `E`. The goal P of symbol `B` is the P that `B` produces a path reaching the goal. In this case, it is the P that it produces `D`, or $1/2$. The P of production `A B−1N1` will be weighted by one minus the goal P and will thus be $1/3 \times 1/2 = 1/6$.

The other six sub-productions will be created by replacing `A` with one of its six sub-symbols and replacing `B` with a sub-symbol that is guaranteed to reach the goal and, when it does, produces a goal whose productions have been filtered by the constraining distribution determined by the `A`'s. The P of each production will be equal to the product of the original production P, $1/3$, the P that `B` reaches the goal, $1/2$, and the source strengths of the sub-`A` symbols used. For example, production `A−1S1.2x1.2 B−1G1.2x1.2` has P $1/3 \times 1/2 \times 1/27 = 0.00617$. Symbol `B−1G1.2x1.2` will be created such that it produces a sub-symbol of `D` whose distribution is filtered by term 1 twice and term 2 once. However, because term 2 eliminates `y`, the term 1 filtering has no effect and this symbol just produces `x`.

The sub-productions for `A A B` will be even more complex because there are two `A`'s. In addition to the production `A A B−1N1` for which `B−1N1` does not reach the goal, we must form a new production for every way that we can replace the `A`'s by sub-`A`'s. In this case, that will result in $1 + 6 \times 6 = 37$ productions. For each, a sub-`B` symbol will be created that is guaranteed to reach the goal and is subject to the appropriate term filters.

After having resolved the constraint and reduced the grammar, the resulting SCFG requires 25 non-terminals and 85 productions. Therefore, assuming that the resulting grammar is what we intended, the use of the constraint reduced the size of the original grammar by a factor of about 6.

B.2.3 Resolving multiple constraints

The constraint resolution process becomes more complicated as we introduce multiple constraints, especially when those constraints share many of the same symbols. Multiple constraints are resolved one at a time. In most cases, the order of resolution does not matter. As we saw in the last section, when a constraint is resolved it generates a number of sub-symbols. When we resolve a second constraint that uses some of those same symbols, the second constraint must be applied to each sub-symbol as it would be applied to the original symbols. As you might well imagine, this has the potential to result in an exponential growth in the number of symbols. Nevertheless, provided that the constraints do not interact too much, fairly large grammars with thousands of constraints can still be resolved. Including the effects of regular expressions, the Penglish grammar has 6,133 constraints.

The resolution process naturally handles many fairly difficult situations that can arise with multiple constraints. For example, one might wonder what happens when constraints are circular. Consider the grammar in Figure B.10. Constraint `AB` says that if `A` produces `ale` then `B` must produce `bed`. Constraint `BC` says that `C` must then be `cat`, which, through constraint `CA` forces `A` to be `awl` and so on. A bit of thought should reveal that the only valid sentence in this language is `ate big cow`, which SLG correctly discovers. If we make the constraints totally circular by adding the term `cow:ale` to function `CA`, SLG will complain that the start symbol is over-constrained and cannot produce anything.

Let us now turn to the problem of resolving two interacting constraints. Consider the grammar in Figure B.11. This contains two constraints, {`AB, A, B`} and {`DC, A D, B C`}, which we will refer to as constraints 1 and 2, respectively. If constraint 1 is resolved first, we are left with the intermediate grammar shown in Figure B.12. We can now resolve constraint 2 as we did in the case of a single constraint, provided we treat the sub-`A` symbols as `A` and the sub-`B` symbols as `B`. `A−1S0.2` does not produce a `D`, so its production will not be affected. `A−1S0.1` only produces

```
S: A B C | {AB, A, B} | {BC, B, C} | {CA, C, A};
A: ale | awl | ate;
B: bed | bus | big;
C: cat | cry | cow;
AB {ale:bed; awl:bus; ate:big;}
BC {bed:cat; bus:cry; big:cow;}
CA {cat:awl; cry:ate;}
```

Figure B.10: A grammar containing circular constraints.

```
S: A B | {AB, A, B} | {DC, A D, B C};
A | B: D | C;
D: w | x;
C: y | z;
AB {D: D (0.2) | C;
    C: D (0.8) | C;}
DC {w: y (0.4) | z;
    x: y (0.6) | z;}
```

Figure B.11: A grammar with two interacting constraints.

D, so we will split it into two sub-symbols, one that produces a sub-D that always produces w and one that produces a sub-D that always produces x. The production A−1S0.1 B−1G0.1 will be split into three sub-productions, each with its own sub-B−1G0.1, as discussed in Section B.2.2.

Figure B.13 shows the final grammar, after resolving constraint 1 followed by constraint 2, in a short-hand notation. Each of the four lines represents one production from the root symbol. Numbers preceding colons are probabilities and brackets represent tree depth. For example, the second line indicates that, with 20% P, a symbol will be produced that produces another symbol that produces a w followed by a symbol that will produce a symbol that produces y 40% of the time and z otherwise.

However, the situation would be more difficult if we had first resolved constraint 2 before constraint 1. Starting with the grammar in Figure B.11 and resolving constraint 2 would have left us in the state shown in Figure B.14. We can begin as usual by creating new symbols along the source path. However, we will not be able to simply create new goal path symbols whose productions reflect the effect of the constraining distribution because each of the sub-B symbols produces either a C or a D. We will not be able to change the relative frequency of C and D simply by modifying the production probabilities of the goal path symbols. In general, problems like these can occur all along the goal path and can be due to the effects of many previous constraints.

To explain how this situation is resolved, we will have to be more explicit about what really goes on in resolving the goal path and root of a constraint. We begin by defining two terms that relate to the sub-symbols that will be created along the goal path. The *goal distribution* of a symbol B on the goal path is the weighted sum of distributions generated by all goal symbols reachable from B. That is, if we start with B and generate all possible ways of traveling down the goal path to the goal (where we might be using sub-symbols created in resolving previous constraints), the goal distribution will be the average of the production distributions of the goal and sub-goal symbols, weighted by the probabilities of reaching those symbols. When we create a new sub-B symbol that is subject to a certain constraining distribution, the new goal distribution of the sub-symbol should be equivalent to the original goal distribution filtered by the constraining distribution. This is true of all symbols on the goal path. The *goal strength* of the sub-B symbol is the dot-product of the original goal distribution and the constraining distribution.

As before, the process of resolving constraint 1 starts by creating the new source path sub-symbols, and creating

```
S: A-1S0.1 B-1G0.1 (0.5) | A-1S0.2 B-1G0.2 (0.5) | {DC, A D, B C};
A-1S0.1: D;
B-1G0.1: D (0.2) | C (0.8);
A-1S0.2: C;
B-1G0.2: D (0.8) | C (0.2);
D: w | x;
C: y | z;
DC {w: y (0.4) | z;
    x: y (0.6) | z;}
```

Figure B.12: The grammar of Figure B.11 after resolving constraint 1.

```
0.5: [C]    [0.8: D | 0.2: C]
0.2: [[w]] [[0.4: y | 0.6: z]]
0.2: [[x]] [[0.6: y | 0.4: z]]
0.1: [D]    [D]
```

Figure B.13: The grammar of Figure B.11 after resolving both constraints, in a short-hand notation.

```
S: A B-1N1 (0.5) | A-1S1 B-1G1 (0.25) | A-1S1.1 B-1G1.1 (0.125) |
   A-1S1.2 B-1G1.2 (0.125) | {AB, A, B};
A | B: D | C;
B-1N1: D;
A-1S1 | B-1G1: C;
A-1S1.1: D-1S0.1;  D-1S0.1: w;
B-1G1.1: C-1G0.1;  C-1G0.1: y (0.4) | z (0.6);
A-1S1.2: D-1S0.2;  D-1S0.2: x;
B-1G1.2: C-1G0.2;  C-1G0.2: y (0.6) | z (0.4);
AB {D: D (0.2) | C;
    C: D (0.8) | C;}
```

Figure B.14: The grammar of Figure B.11 after resolving constraint 2.

sub-productions in the root symbol, `S`. The set of terms that are satisfied by the sub-`A` symbols in each of the new root productions determines the constraining distribution for that production. For each sub-production, we will create a new sub-`B` symbol whose goal distribution has been filtered by the constraining distribution. The new root productions will be as follows:

```
A-2S0.1         B-1N1-2G0.1
A-2S0.2         B-1N1-2G0.2
A-1S1-2S0.2     B-1G1-2G0.2
A-1S1.1-2S0.1   B-1G1.1-2G0.1
A-1S1.2-2S0.1   B-1G1.2-2G0.1
```

If the goal path symbol, `B`, is actually the goal, creating a sub-symbol is easy. We just filter its distribution with the constraining distribution. However, if `B` is not the goal but is further up on the goal path, producing a constrained sub-symbol is more complex. Let's imagine that the symbol following `B` on the source path is `C`. In order to create a sub-`B`, we first recursively create a sub-`C` for each `B` production that uses a `C` and replace the old `C` with the constrained one. The P of the production is scaled by the goal strength of the sub-`C`. Once all productions have been scaled, their probabilities are renormalized as follows. First, each group of productions that derived from the same ancestor production in the original grammar is normalized within itself so that the sum of probabilities in the group remains the same. If any groups were eliminated, all production probabilities are then normalized across the board.

A similar process occurs in the root symbol. Once the new sub-`B` has been created, the P of the new root production using it is scaled by the goal strength of the sub-`B`. When this has been done for each new production, the production probabilities are renormalized within groups, where a group is a set of sub-productions that share the same ancestor in the original grammar and which have the same constraining distribution. If any groups died off because they were over-constrained, the productions are normalized overall.

In the case of our example, production `A B-1N1`, with P = 0.5 was divided into `A-2S0.1 B-1N1-2G0.1` and `A-2S0.2 B-1N1-2G0.2`. The constraining distribution for the former is (0.2:`D` 0.8:`C`) and for the latter is (0.8:`D` 0.2:`C`). The initial goal distribution for `B-1N1` was (1.0:`D` 0.0:`C`). Therefore, the goal strength of `B-1N1-2G0.1` was 0.2 and the goal strength of `B-1N1-2G0.2` was 0.8. As a result, the final P of production `A-2S0.1 B-1N1-2G0.1` is 0.1 and the final P of production `A-2S0.2 B-1N1-2G0.2` is 0.4.

```
0.4: [C]    [D]
0.1: [C]    [C]
0.2: [[w]] [[0.4: y | 0.6: z]]
0.2: [[x]] [[0.6: y | 0.4: z]]
0.1: [D]    [D]
```

Figure B.15: The grammar of Figure B.11 after resolving constraint 2 followed by constraint 1, in short-hand notation.

The resulting grammar, after both constraints have been resolved, is shown in Figure B.15. It is equivalent to the grammar in Figure B.13, which was obtained by resolving the constraints in the opposite order, but does not have exactly the same structure. If the first two productions in Figure B.15 were combined, they would be equivalent to the first production in Figure B.13.

B.2.4 Constraint conflicts

Although most pairs of constraints may be resolved in either order to the same effect, there is one situation in which this is not possible. If the root and goal path of constraint X falls on either the source or goal path of constraint Y, then X must be resolved before Y. Roughly speaking, lower constraints must be resolved before higher ones. The reason is apparent if we consider the example in Figure B.16. We will refer to constraint {foo, C, B} as X and to the other constraint as Y. X's root, A, and its goal path, B, fall on the source path of Y.

```
S: A B | {foo, A B, B};
A: C B | {foo, C, B};
B | C: i | j;
foo {i: i (0.99) | j;
     j: i (0.01) | j;}
```

Figure B.16: A grammar containing a potential constraint conflict.

```
S: A-1S1.1 B-1G0.1 (0.5) | A-1S1.2 B-1G0.2 (0.5);
A-1S1.1: C B-1S0.1 | {foo, C, B};
B-1S0.1: i;
B-1G0.1: i (0.99) | j (0.01);
A-1S1.2: C B-1S0.2 | {foo, C, B};
B-1S0.2: j;
B-1G0.2: i (0.01) | j (0.99);
```

Figure B.17: The grammar of Figure B.16 after constraint Y is resolved.

If we were to resolve Y first, we would be left with the grammar shown in Figure B.17. There are now two sub-A's, each with its own copy of constraint X. But each one produces a sub-B that either produces i or j. We cannot filter the B's goal distributions because they only produce a single symbol. Therefore, it is not possible to resolve X. However, if we were to have resolved X first, it would not have seriously affected the resolution of Y.

Therefore, whenever there is a constraint conflict of this type, the constraints are reordered so the proper constraint is resolved first. However, if the ordering dependencies are circular, there is a problem. SLG gives the user the option of ignoring such conflicts, but a better solution is to restructure the grammar so the constraint ordering is well defined.

B.3 Minimizing the grammar

The process of resolving the grammar creates many new symbols and productions, some of which may be superfluous. Therefore, after resolution, the grammar is minimized to make it more compact. Because of the tradeoff between the number of symbols and the number of productions, there is no clear definition of a minimal CFG, as there is with a finite-state machine. Nevertheless, a number of helpful steps can be taken.

1. **Eliminating dead symbols**. Any symbols which have been so constrained that they have no productions (but are not one of the original terminal symbols) are considered *dead*. Removing a dead symbol requires removing that symbol from all productions using it. If those productions become empty, they too must be eliminated. This can result in another dead symbol. Therefore, eliminating dead symbols is an iterative process.

2. **Eliminating epsilon productions**. The next step is to eliminate any epsilon productions from the grammar. This uses a standard algorithm, described in Hopcroft and Ullman (1979), that has been adapted to properly handle the probabilities in an SCFG. It begins by determining, for each symbol, the P that the symbol produces only epsilon. This uses an iterative procedure that terminates once the values have adequately settled. Ordinarily this only takes a few iterations, but it could potentially settle rather slowly. It might be possible to formulate a closed-form solution to the epsilon probabilities, but this may involve a system of non-linear equations.

   Once the epsilon probabilities have been determined, for each production that uses one or more symbols with a non-zero P of producing epsilon and for each subset of the epsilon-producing symbols in the production, a sub-production is created in which those symbols are eliminated. The P of the sub-production is the product of the original production P, the epsilon probabilities of the symbols that were removed, and the probabilities that the symbols remaining do not reach epsilon.

3. **Combining Equivalent Productions**. This is a relatively simple step in which any pair of identical productions in a symbol is combined into a single production. Also, any productions with 0.0 probability are removed.

4. **Removing Unit Productions**. A *unit production* is a production that contains just one non-terminal. That is, one non-terminal is simply replaced by another one. As shown in Hopcroft and Ullman (1979), if a grammar uses unit productions, there is always an equivalent grammar that does not. Because this process can change the structure of the grammar, it is only done when *aggressive* minimization is requested. Although Hopcroft and Ullman (1979) mention an algorithm for removing unit productions in a CFG, it is not efficient for an SCFG.

   The algorithm used in SLG iterates over the non-terminal symbols. For each symbol, `A`, it first removes any self-unit productions, which are always unnecessary, and renormalizes the remaining productions in `A`. It then searches in other symbols for any unit productions that use `A`. These productions are removed and replaced with `A`'s productions, with their probabilities scaled by the P of the original production. Any newly created equivalent productions or self-productions are then removed. When this process is completed, all unit productions will have been removed.

5. **Removing Equivalent Symbols**. The next step in minimization is to remove any symbols that have identical sets of productions. The reduction process tends to create a lot of these. In order to do this efficiently, the symbols are first sorted based on their productions. Then neighboring symbols are compared and duplicates removed. Any references to the duplicate symbols within productions must be changed to refer instead to the surviving symbol. Unless aggressive minimization is requested, two symbols which were generated from different ancestor symbols during the reduction process are not considered equivalent.

   Because replacing equivalent symbols can create equivalent productions, Step 2 must be repeated. This, in turn, can create more equivalent symbols so Step 4 is run again. This continues until there are no more equivalent symbols. This usually takes just a few iterations.

6. **Removing Unreachable Symbols**. Finally, any symbols that are not reachable from the start symbol, and could therefore never participate in the grammar, are removed.

The constraint resolution process tends to produce a lot of equivalent symbols and some dead ones. Therefore, after each constraint is resolved, any dead symbols are removed and equivalent symbols that share the same ancestor are consolidated. For a grammar like the one used in the Penglish language, this considerably reduces the memory required for the resolution.

B.4 Parsing

Aside from generating sentences using the grammar, some users may wish to perform the reverse process of parsing sentences into their syntactic structures. One efficient and easily implemented parsing method is the CYK algorithm (Hopcroft & Ullman, 1979), which is based on dynamic programming. However, the CYK algorithm requires that the grammar be in a particular format in which each production has at most two symbols. This is known as Chomsky normal form (CNF). The one stipulation in implementing this algorithm in SLG is that when we parse we would like to obtain the syntactic structure of the sentence in terms of the original CFG grammar, not in terms of the CNF version of the grammar. Therefore, SLG uses a slightly loosened definition of CNF.[2] Section B.4.1 explains how an SCFG can be converted to CNF without losing track of the original structure of the grammar.

The CYK algorithm, as implemented in SLG, constructs a two-dimensional table, P. Each entry in the table is associated with a substring of the sentence to be parsed. Any substring can be identified by specifying the position of the first word in the substring and the length of the substring, (i, j). Likewise, entry $P[i][j]$ corresponds to the

[2]The definition of Chomsky normal form used here allows a production to contain a single non-terminal, while the standard definition is that each production consist of either two non-terminals or one terminal.

substring, (i, j), that starts with word i and is j words long. When entry $P[i][j]$ is completed, it will contain a list of all symbols whose full expansion could result in the production of substring (i, j).

The algorithm begins by filling in all entries of length 1, $P[i][1]$. These represent substrings consisting of single words, so parsing them is not too hard. The one problem to consider is that a terminal symbol could be the sole product of a non-terminal symbol. Such a production could even go through a chain of non-terminals and any non-terminal that can produce that word by itself must be added to entry $P[i][j]$. Therefore, filling in an entry involves an iterative process to find all non-terminals that could generate just the word in question. The reason the standard definition of CNF prohibits productions consisting of a single non-terminal is to avoid this iterative process.

The algorithm then fills in all entries of length 2, then 3, and so on. When filling entry $P[i][j]$, we know that entries $P[r][s]$ have already been completed, for all $s < j$. Any symbol that produces substring (i, j) must either have a one-symbol production that generates another symbol that produces (i, j) or it must have a two-symbol production, the first symbol of which produces (i, k) and the other symbol of which produces $(i + k, j - k)$, where $0 < k < j$. That is, the first symbol produces part of the substring and the other symbol produces the rest of the substring. In order to fill in entry $P[i][j]$, we must find all symbols for which this is true.

To find all symbols that can produce $P[i][j]$, we iterate over k. At each step, we then iterate over all pairs of symbols from lists $P[i][k]$ and $P[i + k][j - k]$, respectively. For each pair of symbols, `A B`, we find all other symbols with production `A B` and add those symbols to list $P[i][j]$. When we add those symbols, we also maintain a record of the fact that production `A B` was used to generate substring (i, j), and k was the division point. That way, we can later reconstruct the complete parse tree. In order to efficiently find all matching symbols, a hash table of productions is maintained. By looking up `A B` in the hash table, we obtain a list of all symbols with production `A B`.

The algorithm terminates when entry $P[1][n]$ has been computed, where n is the number of words in the sentence. Any non-start symbols in the entry can be discarded, because sentence production has to begin with a start symbol. If there is no start symbol in the entry, the sentence has no parse. If there are multiple start symbols, each will be associated with a different parse of the sentence. This algorithm finds all possible parses.

Having chosen one of the parses, we can generate the complete parse tree by working our way back through the table. If we want the parse tree to reflect the original form of the grammar, and not the CNF grammar, we simply ignore all symbols that were created in the transition to CNF.

B.4.1 Converting an SCFG to Chomsky normal form

Converting an SCFG to the relaxed CNF used by SLG is quite easy. Any production containing one or two symbols is left as is. If a production, Q, has three symbols, `A B C`, a new symbol `#1` is created with the single production `B C` and Q is replaced by `A #1`. If a production, Q, has more than three symbols, it is replaced by one containing two new symbols, `#1 #2`. `#1` will have a single production generating the first half of Q and `#2` will have a single production generating the second half of Q. As the process continues, these productions will eventually be split if they contain more than two symbols. The newly created symbols can easily be ignored in producing the parse trees after running the CYK algorithm because they all begin with a `#`.

B.5 Word prediction

Although the CNF is an efficient representation for parsing whole sentences, it is not well suited to performing word prediction given part of a sentence, particularly if we would like to do it iteratively after each successive word.

The method used by SLG to perform prediction relies on a grammar in Greibach normal form, in which each production must begin with a terminal. With the grammar in this form, word prediction is relatively easy. As the sentence is processed, from left to right, the parser keeps a list of every possible continuation with their associated probabilities. Continuations are in the form of a terminal followed by one or more other symbols that could produce the remainder of the sentence. Given a set of continuations, it is easy to generate a distribution of next words because the continuations begin with terminal symbols.

When the next word is processed, continuations not starting with that word are discarded. The first word is dropped from each remaining continuation and, if the new first symbol is a non-terminal, new continuations are created with the first symbol replaced by each of its productions. While, in theory, this algorithm could generate an exponentially large list of continuations for a highly ambiguous grammar, in practice it does quite well on pseudo-natural languages. Natural languages tend to be only mildly ambiguous, especially if semantic constraints are enforced. Most ambiguities are only temporary and are resolved quickly. If natural languages were too ambiguous, humans would not be able to parse them either.

B.5.1 Converting an SCFG to Greibach normal form

In order to convert an SCFG to Greibach normal form (GNF), the algorithm described in Hopcroft and Ullman (1979) was adapted to handle production probabilities. The algorithm need not start with a grammar in Chomsky normal form, but we will relax the restriction that all symbols following the first terminal in a GNF production must be non-terminals. Figure B.18 shows a modified version of the first step of the algorithm, indicating how the probabilities of new productions should be calculated to maintain equivalence. Probabilities are listed in parentheses following each production. The notation P_Q refers to the P of production Q being generated by its parent symbol.

for $k \leftarrow 1$ **to** m **do**
    for $j \leftarrow 1$ **to** $k-1$ **do**
        for each production, Q, of the form $A_k \rightarrow A_j\alpha$ **do**
            for each production, R, of the form $A_j \rightarrow \beta$ **do**
                add production $A_k \rightarrow \beta\alpha\ \ (P_Q \times P_R)$
            remove production $A_k \rightarrow A_j\alpha$
    $x \leftarrow 0$
    for each production, Q, of the form $A_k \rightarrow A_k\alpha$ **do**
        $x \leftarrow x + P_Q$
    for each production, Q, of the form $A_k \rightarrow A_k\alpha$ **do**
        add production $B_k \rightarrow \alpha\ \ (P_Q \times (1-x)/x)$
        add production $B_k \rightarrow \alpha B_k\ \ (P_Q)$
        remove production $A_k \rightarrow A_k\alpha$
    for each production, Q, of the form $A_k \rightarrow \beta$, where β doesn't begin with A_k **do**
        add production $A_k \rightarrow \beta B_k\ \ (P_Q \times x/(1-x))$

Figure B.18: A modified version of Figure 4.9 of Hopcroft and Ullman (1979) indicating how to handle production probabilities in converting to GNF.

Once this step is complete, it will be the case that, for all productions, Q, of the form $A_i \rightarrow A_j\gamma$, i will be less than j. Therefore, we can eliminate all such productions by replacing them with all productions formed by replacing A_j with one of its productions, R. The P of the new production will be $P_Q \times P_R$. As long as we start with the last symbol and work to the first, we will never introduce a new production that starts with a non-terminal. A similar process can then be performed to replace all productions of the form $B_i \rightarrow A_j\gamma$. Because there can be no $B_i \rightarrow B_j\gamma$ productions, all productions will now begin with a terminal symbol.

B.6 Conclusion

SLG is intended to help users design interesting context-free languages. It is especially useful in creating training environments for machine-learning experiments. Without using constraints, it is still a convenient tool for working with stochastic context-free and regular languages. But the use of constraints can greatly simplify the writing of pseudo-natural languages by separating the syntax of the underlying grammar from semantic and pragmatic influences and allowing important contingencies to be carefully controlled.

Although the method for specifying SLG constraints is quite powerful, it does have some limitations and there are several possible extensions that may improve it. Currently, all of the constraints for a particular root symbol must be satisfied. Therefore they essentially form a logical conjunction. The grammar could be more flexible if it allowed an arbitrary Boolean formula of constraints to be specified for each symbol. For example, one might specify that either constraint A and constraint B must apply or constraint C may apply. If we had a language with adjectives and compound nouns, we might wish to produce phrases such as "`the happy dog and the sad dog`" or "`the happy dog and the happy boy`", but not "`the happy dog and the happy dog`", which would be redundant. We could do this by specifying that either the nouns must differ or the adjectives must differ. In the current implementation, this is possible, but much less convenient.

Another shorthand that may be useful is the addition of single-path constraints. That is, one might filter the productions of a goal symbol at the end of a particular path out of the root symbol, but not in a way contingent on context. These would be helpful in simplifying many grammars and could be used to bound the depth of recursion.

SLG is written in C and should compile on most systems. The source for the latest version is available at...

http://www.cs.cmu.edu/~dr/SLG

Appendix C

TGREP2: A Tool for Searching Parsed Corpora

This appendix explains the TGREP2 program which, in conjunction with its predecessor TGREP, was used in conducting the corpus analysis (see Section 4) that provided the statistical information used in designing the Penglish language.

Statistical analysis of large, syntactically tagged corpora is playing an increasingly important role in language research. In particular, the Penn Treebank (Marcus et al., 1993, 1994) version of the Wall Street Journal and Brown corpora of English text is a frequently studied resource. Until now, the software tool of choice for analyzing such corpora has been the TGREP program, developed by Richard Pito.

The primary function of TGREP is to extract parse trees whose structures match a specified pattern. It is, essentially, *grep* for trees.[1]However, working with TGREP and talking to others who use it in their research resulted in a wish list of improvements. Eventually, I undertook to rewrite TGREP in its entirety. The result is TGREP2. TGREP2 is almost completely backward compatible with TGREP, but introduces a number of new features, including the following major enhancements:

- Rather than simply having a set of required relationships and a set of prohibited relationships, nodes can have full Boolean expressions of relationships to other nodes.
- Nodes can be given unique labels and may then be referred to by those labels in the pattern specification or in selecting trees for printing.
- Patterns are no longer restricted to simple tree architectures. The use of node labels and segmented patterns allows links in a pattern to form back-edges as well, permitting cycles of links.
- Customizable output formats allow a variety of information to be reported in a flexible manner.
- Multiple search patterns may be specified and the user can retrieve the first subtree matching any pattern, the first subtree matching each pattern, or all subtrees matching all patterns.
- Subtrees can be reported using a code rather than by printing the whole structure. The trees themselves can later be retrieved using the codes.
- A variety of new links have been added and the immediately-precedes link now has a more conventional meaning.
- TGREP2 corpus files are substantially smaller than TGREP corpora.

C.1 Preparing corpora

Before using TGREP2, a special corpus file must be created in a format that is a bit different from that of the corpus files used by TGREP. Corpus files are generated using the **-p** option to TGREP2. Following the **-p** are two arguments,

[1]Grep, which stands for "global regular expression print," is a standard Unix tool for retrieving lines in a text file that match a given pattern.

one giving the name of a text input file containing the corpus trees and one giving the name of the binary output file to which the corpus will be written. TGREP2 corpus files normally end in the ".t2c" extension, but that is not required.

The input file contains sentences in parenthesized tree format.[2] Each tree must begin with an open parenthesis. Here is an example of an input file with three very short sentences:

```
(TOP (NP (NP (NN Research)) (CC and) (NP (NN development))))
(TOP (NP (NP (NN Budget)) (VP (VBD increased))))
# This is a comment.
(TOP (NP (JJ Stretch) (NN yarn) (NNS machines)))
```

There is currently a limit of 255 children per node and 65,535 nodes per sentence due to the format in which corpus files are stored. The limit on children per node can be raised to 65,535 by using the **-K** flag when building the corpus.

Comments can appear in the input file if they are on a line starting with a `#`. By default, comments in the input file are not stored in the corpus file. But if the **-C** option is specified prior to **-p**, the comment immediately preceding each sentence will be associated with the sentence and recorded in the corpus. The comments can later be printed when a match is made during a search.

If either the input or output files have the .gz or .Z extension, they will automatically be decompressed or compressed, respectively. If the input file is compressed, the .gz or .Z extension need not be specified. To write to standard output, "**–**" can be given in place of the output file. However, the input file cannot be read from a pipe or standard input. The input must come from a static file.

Here is one procedure for converting a TGREP corpus file to a TGREP2 corpus file:

```
% export TGREP_CORPUS=wsj_mrg.crp
% tgrep -n __ | grep . | gzip > wsj_mrg.txt.gz
% tgrep2 -C -p wsj_mrg.txt wsj_mrg.t2c.gz
```

On most machines, it is not much slower (and is sometimes faster) to use a compressed corpus file rather than an uncompressed one because of the tradeoff between the decompression time and the reduced disk access. When uncompressed, TGREP2 corpus files are about 20% the size of TGREP corpus files. And when compressed, they are under 5% the size of an uncompressed TGREP corpus.

C.2 Command-line arguments

TGREP2 is used as follows:

```
% tgrep2 [options] <pattern>
```

The final argument is normally either a pattern or the name of a file containing the pattern. If there is a readable file matching the final argument, the pattern will be taken from that file. Otherwise, the final argument will be treated as the pattern itself. But there are two exceptions to this rule. If the **-e** option is used, a filename containing subtree codes is used in place of the pattern. If the **-p** option is used, the final pattern is optional.

See Section C.3 for more about specifying search patterns.

C.2.1 Options

-c <corpus-file> This sets the corpus file used by TGREP2. If no corpus is specified on the command line, it must be given by the **TGREP2_CORPUS** environment variable.

[2] Although the ability to search for tree structures may be helpful in many other domains, TGREP2 is most often used for processing corpora of parsed sentences. Therefore, the terms *tree* and *sentence* are used interchangeably here, since each sentence is associated with a single parse tree.

-a Ordinarily, only the first subtree matching each pattern is reported. This option causes *all* subtrees matching each pattern to be reported. Note that a pattern might match a single subtree in more than one way. In this case, only the first match between pattern and subtree is reported, even with the **-a** option. This option is not compatible with **-f**.

-f Ordinarily, only the first subtree matching *each* pattern is reported. This causes only the first subtree matching *any* pattern to be reported. If the first pattern matches a subtree, the second pattern will not be considered. This option is not compatible with **-a**.

-v Normally, TGREP2 reports subtrees that do match the pattern(s). This option invokes "filter mode," which causes trees to be reported which *do not* match any patterns. If the output format is specified with **-m**, it will behave as if a match occurred on the top node in the tree.

-b <format> This specifies the output format that determines what is printed at the start of every sentence in the corpus, whether or not there is a match. See Section C.4.2 for details on formatted output.

-s <format> This specifies the output format that determines what is printed before the first match on sentences for which there is at least one match. If the **-v** option is used, this defines the format used in printing sentences with no matches. If the **-f** option is used, this is essentially redundant with **-m**. See Section C.4.2 for details on formatted output.

-m <format> This specifies the output format that determines what is printed every time there is a match between a pattern and a sentence subtree. If this is not defined, then matching subtrees are printed one-per-line in short (parenthesized) form, unless either **-l**, **-t**, **-u**, or **-x** is given. See Section C.4.2 for details on formatted output.

-l If **-m** is not specified, this causes matching subtrees to be printed in long format (indented with one symbol per line).

-t If **-m** is not specified, this causes only the terminals of matching subtrees to be printed. Thus, the trees are printed like a sentence, with no structural information.

-u If **-m** is not specified, this causes only the name of the top symbol of each matching subtree to be printed. This will print the name of a non-terminal without printing its children.

-x If **-m** is not specified, this prints only the subtree code for matching subtrees. This is a unique identifier for the subtree and is in the form $s{:}n$, where s is an integer specifying the sentence number in the corpus (starting with 1), and n is an integer giving the order in which the node is encountered in a depth-first search starting with 1 at top node in the sentence tree.

-w If **-m** is not specified, the whole tree is printed on each match, rather than the matching subtree. The format in which the tree is printed is affected by the use of the **-l** and **-x** flags.

-d Ordinarily, TGREP2 reorders the links in the search pattern for greater efficiency. This can alter the order in which marked nodes are printed and the order in which links in a disjunction are searched. If that is a problem, the **-d** option will suppress the reordering of links.

-r <seconds> If a positive integer is specified, it determines how often, in seconds, a progress report is printed. Progress reports consist of the completion percentage and are written to the standard error output. By default, there are no progress reports.

-e This option causes TGREP2 to operate in a special "extraction mode". The final command-line argument should be the name of a file containing subtree codes, rather than a pattern. A subtree code is in the form $s{:}n$, where s is an integer specifying the sentence number in the corpus (starting with 1), and n is an integer giving the order in which the node is encountered in a depth-first search starting with 1 at top node in the sentence tree.

Each subtree whose code is given will be reported. For the purposes of formatted output, it behaves as if the head node in the pattern matched the indicated subtree. The subtree codes must be ordered so that the sentence numbers are non-decreasing.

-p <text-file> <corpus-file> This is used to generate a TGREP2 corpus file. See Section C.1 for details on corpus files. When using this option, if no search pattern is given, TGREP2 will simply exit. But one can also specify a pattern to be searched once the corpus is built. This makes it possible to generate a corpus and search the corpus in a single step. To facilitate this, the active corpus is set to the one newly built. This can be overridden with a subsequent use of **-c**.

-z This causes TGREP2 to parse the patterns, possibly reorder the links for improved speed, and pretty-print the patterns back to standard output before exiting. This is helpful for debugging complex patterns.

-C If this precedes **-p**, comments in the input file will be stored in the corpus. Multiple lines starting with `#` will be considered one comment if they are consecutive, but separate comments if there is a blank line between them. Only the comment immediately preceding a sentence is stored.

This option can also be used to print the comments while searching a corpus. If you are not using formatted output (not using the **-m** flag), any comment associated with a sentence will be printed just before the first match made to that sentence.

-K Ordinarily there is a limit of 255 children per parse tree node. That is because the number of children is stored using an 8-bit char in the corpus file. If you have nodes with more than 255 children, you can use the **-K** flag prior to the **-p** flag to increase the limit to 65,535. This causes the number of children to be stored using 16-bit words, so the corpus file will be a bit larger.

-h Prints a summary of the command-line arguments, output formatting fields, and link codes.

C.3 Specifying patterns

Unless the **-e** or **-p** options are used, the last argument to TGREP2 must be either a pattern or the name of a pattern file. If the argument happens to be the name of a readable file, the file will be used. Otherwise, it will be treated as the pattern itself.

It is generally a good idea to store all patterns in files, rather than always using the command line. Complex patterns can be too long and unwieldy for the command line. Also, a pattern used now will eventually be needed again and it is smart to have a record of it.

The format for patterns is the same whether they are in a file or on the command line. Line breaks in the file are simply treated as white space. However, any line starting with a hash mark (`#`) is considered a comment and ignored. If multiple patterns are specified, it is not sufficient to put them on different lines. They must be separated with semicolons (`;`), as discussed in Section C.3.7.

C.3.1 Basic pattern syntax

TGREP patterns ought to work just fine with TGREP2, but TGREP2 patterns can be significantly more expressive.

TGREP2 patterns mainly consist of node names and relationships, which define *links* to other nodes. A simple node consists of just a node name, which is a string, regular expression, or OR'd combination of the two that is matched against the names of nodes in the sentence tree. A complex node consists of a node name followed by a set of relationships, all surrounded by parentheses.

The first node in the pattern is called the *head* node. It is not necessary to enclose the head node and its relationships in parentheses.

As in TGREP, a set of relationships can simply be a list of links or negated links to other nodes, all of which must be satisfied for the node to match a subtree. However, TGREP2 also allows Boolean expressions of relationships, as explained in Section C.3.4.

Nodes can also be assigned labels and may be referred to elsewhere in the pattern by those labels (see Section C.3.5).

Finally patterns can be broken into multiple segments (see Section C.3.6) and multiple patterns may be specified (see Section C.3.7).

Whenever a new pattern is written, it is a good idea to check the pattern by running TGREP2 with the **-z** option. This parses the pattern and reprints it in standard format before exiting and is helpful in diagnosing problems that do not result in syntax errors. Note that when a pattern is parsed, segments are removed, crossing-links are replaced by copies of the nodes to which they point, and the links are reordered for improved efficiency. The **-d** option can be used to prevent link reordering.

By and large, spaces are optional in patterns. But for readability, it is usually best to place spaces between link codes, node names, and other operators.

Example patterns

This matches any NP node that immediately dominates a PP:

```
NP < PP
```

This is a regular expression that matches any node whose name starts with NP, including NP-SBJ:

```
/^NP/
```

This matches an NP that dominates a PP *and* is immediately followed by a VP:

```
NP << PP . VP
```

This matches an NP that dominates a PP *or* is immediately followed by a VP:

```
NP << PP | . VP
```

This matches an NP that does not dominate a PP. Also, the NP must either have a parent that is an NP or be dominated by a VP:

```
NP !<< PP [> NP | >> VP]
```

This matches an NP that dominates a PP which itself is immediately followed by a VP. Note the use of parentheses to group "`. VP`" with the PP rather than with the NP:

```
NP << (PP . VP)
```

This matches an NP whose last child is a PP that begins with the preposition on:

```
NP <` (PP <, (IN < on))
```

Labels allow the creation of patterns with cycles in the link structure, which is often quite useful. For each NP that is followed by a VP, the following pattern prints the lowest node dominating both the NP and the VP (when used with the **-a** option). Note that the NP is given the label n and is later referred to by that label. * matches any node and is thus equivalent to __. The ` marks that node for printing:

```
NP=n .. (VP=v >> (`* << =n))
```

The following pattern is a bit different. It matches any node that dominates an NP and a VP, such that the NP is followed by a VP and the NP and VP are not dominated by a node lower than the original one. This differs from the previous pattern in that the prior one may print the same subtree more that once.

```
*=p << (NP=n .. (VP=v >> =p !>> (* << =n >> =p)))
```

As one can see, patterns can become rather complex when using labels. However, judicious use of labels and Boolean expressions can greatly simplify searches. This is discussed more in Sections C.3.4 and C.3.5.

C.3.2 Node names

A node name is an expression that matches the name of a single node in the tree. Node names are similar in TGREP and TGREP2. A node name can be a simple constant string (e.g., NP-SBJ), a regular expression (e.g., /^NP/), or any number of these things separated by pipes (|) with no spaces (e.g., S|/^SBAR/).

A constant string that is not enclosed in double quotes cannot contain white space or any of the following special characters: ;:.,&|<>()[]$!@%`^=

If a constant string is enclosed in double quotes, it can contain any symbols (e.g., "*$#&%!"). If it must contain a double quote, the double quote should be preceded by a backslash.

Regular expressions must be enclosed in forward slashes. See the *grep* manual page for an explanation of regular expression syntax. As with TGREP, the pattern "__" matches any node. In TGREP2, the more conventional "*" is preferred.

The following pattern matches names like Robert, Bob, bob, bobby, and Bobaflooper:

```
Robert|/^[Bb]ob/
```

C.3.3 Basic links

Relationships define connections between the node being defined and other nodes. A relationship consists of a link followed by a node. The links used in TGREP have been adopted and some new links have been added so there is a complete pairing of forward and backward links.

A < B	A is the parent of (immediately dominates) B.
A > B	A is the child of B.
A <N B	B is the Nth child of A (the first child is <1).
A >N B	A is the Nth child of B (the first child is >1).
A <, B	Synonymous with A <1 B.
A >, B	Synonymous with A >1 B.
A <-N B	B is the Nth-to-last child of A (the last child is <-1).
A >-N B	A is the Nth-to-last child of B (the last child is >-1).
A <- B	B is the last child of A (synonymous with A <-1 B).
A >- B	A is the last child of B (synonymous with A >-1 B).
A <` B	B is the last child of A (also synonymous with A <-1 B).
A >` B	A is the last child of B (also synonymous with A >-1 B).
A << B	A dominates B (A is an ancestor of B).
A >> B	A is dominated by B (A is a descendant of B).
A <<, B	B is a left-most descendant of A.
A >>, B	A is a left-most descendant of B.
A <<` B	B is a right-most descendant of A.
A >>` B	A is a right-most descendant of B.
A . B	A immediately precedes B.
A , B	A immediately follows B.
A .. B	A precedes B.
A ,, B	A follows B.
A $ B	A is a sister of B (and A $\neq$ B).

`A $. B`	`A` is a sister of and immediately precedes `B`.
`A $, B`	`A` is a sister of and immediately follows `B`.
`A $.. B`	`A` is a sister of and precedes `B`.
`A $,, B`	`A` is a sister of and follows `B`.

Note that the immediate precedence relationship in TGREP2 works as one might expect, rather than the odd way it was defined in TGREP. Tree node `A` immediately precedes node `B` if the last terminal symbol (word) produced by `A` immediately precedes the first terminal symbol produced by `B`. In TGREP, `A` only immediately preceded `B` if `B` was the next node in depth-first search order after `A` that was not a descendant of `A`, which is more restrictive.

An exclamation mark (`!`) can be placed immediately before any link to negate it. Thus, `A !.. B` means that `A` is not followed by `B`. This is actually a special case of negation in the more general Boolean expressions described in the next section.

In order to be compatible with TGREP, several alternative characters are permitted as replacements in specifying links, but their use is discouraged:

- `!` can be replaced by `@`
- < can be replaced by { or by ^
- > can be replaced by }
- `$` can be replaced by `%`

C.3.4 Boolean expressions

One of the major additions in TGREP2 is the ability to specify Boolean expressions of relationships. Formerly, for each node in the pattern, one could only specify a set of relationships that must exist and a set of relationships that could not exist (by using the `!` operator). What was lacking, essentially, was disjunction.

In TGREP2, as in TGREP, if no operator is placed between two relationships, they are assumed to be connected by an *and*. In TGREP2 an ampersand (`&`) can optionally be placed between them for clarity. The following patterns both mean that `A` has child `B` *and* is immediately followed by `C`:

```
A < B . C
A < B & . C
```

If a pipe (|) is placed between two relationships, only one of them must be satisfied. The possible relationships are tested in order. If one matches, the remaining ones are ignored. This means that node `A` has child `B` *or* is immediately followed by `C`:

```
A < B | . C
```

The *and* binds tighter than the *or*. So this expression evaluates to (`A` has child `B` *and* is immediately followed by `C`) *or* (`A` has child `D` *and* is immediately followed by `E`):

```
A < B . C | < D . E
```

In order to create more complex expressions, square brackets (`[]`) can be used to group terms. Thus, (`A` has child `B` *or* is immediately followed by `C`) *and* (`A` has child `D` *or* is immediately followed by `E`), can be written:

```
A [< B | . C] [< D | . E]
```

The bracketed expression itself acts, syntactically, as a relationship. Note that an open bracket never follows a link but always precedes a link or another open bracket.

Any relationship, including a bracketed one, can be negated by immediately preceding it with an exclamation mark (`!`). Thus, this expression means (`A` has child `B` *or* it does *not* (immediately precede `C` and *not* immediately follow `F`)) *or* (`A` does *not* (have child `D` *and* is *not* followed by `E`)):

```
A [< B | ![. C !, F]] | ![< D !.. E]
```

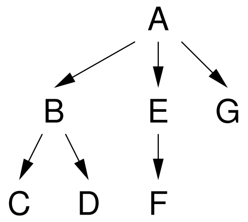


Figure C.1: The link structure of pattern "`A .. (B !< C . D) | ![<< (E , F) $ G]`"

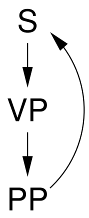


Figure C.2: The link structure of pattern "`S=foo << (VP .. (PP >> =foo))`"

C.3.5 Labeled nodes

Another major addition of TGREP2 is that a node can be given a unique label and then referred to elsewhere by that label. To explain why that is useful, we will need a bit of notation.

It is helpful to think of a TGREP2 pattern as a graph. Nodes are vertices and there is an edge from node `A` to node `B` if there is any link from `A` to `B` in the pattern. Thus, the pattern:

```
A .. (B !< C . D) | ![<< (E , F) $ G]
```

has the graph structure shown in Figure C.1.

Without the ability to label nodes and refer to them by label, it is not possible to express patterns whose links do not have a regular tree structure. Using labels permits the construction of patterns with more complex structures, which can make patterns more concise and allow one to perform some queries that are not possible otherwise.

For example, imagine that we wanted to find a sentence node, `S`, that dominates both a `VP` and a `PP` such that the `VP` precedes the `PP`. This cannot be expressed in a tree of links because there is a relationship between the `S` and the `VP`, between the `S` and the `PP`, and between the `VP` and the `PP`. However, it can be written in TGREP2 as follows:

```
S=foo << (VP .. (PP >> =foo))
```

`S=foo` matches any tree node whose name is `S`. Furthermore, when a matching tree node is found, it is given the label `foo`. Later, "`PP >> =foo`" indicates that the `PP` must be dominated by that very same node, not just any `S`. The tree structure of this pattern is shown in Figure C.2.

If a node name is followed by an equal sign (=) and a label, the label will be assigned to the node. If the node name only consists of an equal sign and a label, it refers to the node that has been assigned to that name elsewhere in the pattern. Every label must be assigned to exactly one node. Labels may not contain any of the special characters that are prohibited in node names unprotected by double quotes.

The way that a node label is used is actually a bit complex and can be confusing. If node `A` has a link that refers to the node labeled `foo` and the label `foo` is assigned to node `B`, the meaning of that link depends on the relationship between `A` and `B`.

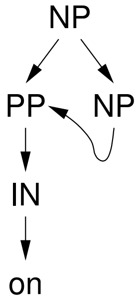
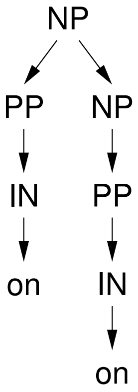


Figure C.3: The initial structure of pattern “`NP < (PP=pp < (IN < on)) | < (NP < =pp)`”



1

Figure C.4: The final structure of pattern “`NP < (PP=pp < (IN < on)) | < (NP < =pp)`”

If `B` is an ancestor of `A` in the pattern tree (as in the previous example), the link forms a *back link*. In this case, the label refers to the same tree node matched by `B`. Thus, the same node is the ancestor of both the `VP` and the `PP`.

However, it is possible to express patterns in which the labeled node `B` may not be an ancestor of `A`. Consider this example:

```
NP < (PP=pp < (IN < on)) | < (NP < =pp)
```

In this case, the `NP` has a link to the node labeled `pp`, but the node labeled `pp` is not an ancestor of the `NP`. If we were to draw the link structure of this pattern, it would look like Figure C.3.

The strange link in Figure C.3 is a *crossing link*. A crossing link extends from a node to a subtree on its left. The problem with crossing links is that they can make matching a pattern to a tree an extremely slow and complicated computational process. Therefore, TGREP2 does not permit crossing links. If a link to a labeled node forms a crossing link, then a copy is made of the whole subtree in the pattern pointed to by the link. The result is shown in Figure C.4.

In this case, the use of a node label didn't enable us to do something we couldn't have done without a label. It simply made the pattern more concise. Note that if a node is copied and has been assigned a label, the new node will have the same label with one or more pluses (`+`) appended to the end to create a unique label.

A final convenience of node labels is that they can be referred to when producing formatted output. A subtree can be printed by specifying the label of the node matching that subtree. See Section C.4.2 for more about formatted output.

C.3.6 Segmented patterns

For clarity's sake, it is not always convenient to write a complex pattern in fully parenthesized form. It is easy to get lost in a sea of nested parentheses. Therefore, TGREP2 provides a way to split patterns into multiple segments, which are separated by colons (`:`). Each segment following the first must begin with a reference to a labeled node that is defined in a previous segment.

In fact, if all complex nodes are labeled, segments can entirely replace parentheses. Consider this example, which expresses the same pattern with and without segments:

```
S < (NP=n1 .. (VP=v < PP)) < (NP=n2 !.. VP)
(S < NP=n1 < NP=n2):(=n1 .. VP=v):(=v < PP):(=n2 !.. VP)
```

The segments do not necessarily make the pattern shorter, but they can make it easier to read and debug. If patterns are stored in files, it may be convenient to place each segment on its own line.

It is frequently the case that one compares several patterns that are mostly the same but differ in only one region. If the region that changes is placed in its own segment, then it is easy to compose the full patterns by appending a unique segment to a common base. If a change is made in the base, it will propagate to all of the patterns built from that base.

C.3.7 Multiple patterns

It is sometimes necessary to compare or combine the results of multiple search patterns. With TGREP, one must run the program separately for each pattern. However, TGREP2 provides a mechanism for specifying an arbitrary number of search patterns in a single invocation.

Multiple patterns are separated by semicolons (`;`). The patterns are independent of one another. Node labels used in one pattern can only refer to nodes in that pattern and are disjoint from labels used in other patterns. In this example, one pattern selects `NP`s containing `PP`s and the other pattern selects `NP`s that do not contain `PP`s:

```
NP << PP; NP !<< PP
```

Ordinarily, TGREP2 tries to match the first pattern to a subtree. If a match is found, it is printed and the next pattern is matched against the tree. If there are N patterns, there can be up to N matches. If the **-f** option is used, only the first match to any pattern is printed. So there can be at most one match to any sentence.

If the **-a** option is used, all subtrees matching each pattern are printed. Note that this does not mean that every possible way of aligning the pattern nodes to trees are considered matches. Different matches must assign the head node of the pattern to different tree nodes. Thus the pattern, "`S << NP`" will match a particular `S` only once, no matter how many `NP`s it contains.

When using formatted output, one has the option of printing the number of the pattern resulting in the match (**%p**) and the number of times that pattern has resulted in a match on the current sentence (**%j**). This makes it possible to distinguish subtrees matched by the various patterns in the output.

C.4 Controlling the output

By specifying output formats, it is possible to have TGREP2 report a variety of information in a customizable format. This is detailed in Section C.4.2. But for most users, the simpler command-line arguments **-w**, **-l**, **-t**, **-u**, and **-x** will suffice.

By default, when a pattern matches a tree in the corpus, TGREP2 prints the subtree matched by the head node in the pattern. The head node is the first node specified. If the **-w** option is used, the whole sentence is printed, regardless of where within it the match occurred.

The default format for printing trees is the *short form*. This renders each tree on a single line with non-terminals enclosed in parentheses:

```
(NP (NP (NNS sales)) (PP (IN of) (NP (NNP U.S.) (NNS savings) (NNS bonds))))
```

The **-l** option causes the *long form* to be used, which renders trees across multiple lines. This was the default in TGREP:

```
(NP (NP (NNS sales))
    (PP (IN of)
        (NP (NNP U.S.)
            (NNS savings)
            (NNS bonds))))
```

The **-t** option prints only the terminals (words) of the sentence, eliminating all of the structural information:

```
sales of U.S. savings bonds
```

The **-u** option prints only the name of the top symbol in each matching subtree. It will not print the children of that symbol. This would normally be used when the symbol is matched by a wildcard or regular expression, and thus could have several possible names.

Finally, the **-x** option prints the *subtree code* for each match. The subtree code consists of the sentence number followed by a colon (`:`) and the number of the top node in the subtree in the order in which it is encountered on a depth-first search. See Section C.4.3 for more information on using subtree codes to later retrieve those subtrees from the corpus.

If you ever request that a node be printed, but the node is not matched to a subtree in the current tree, "`<none>`" is printed in place of the tree. However, if the **-x** option is in use, the sentence number will be printed followed by a colon and 0 for the subtree number.

C.4.1 Marking nodes for printing

As in TGREP, it is possible to select nodes other than the head node for printing. If the name of any node is preceded by a single right-quote (`), the tree matched by that node will be printed, rather than the head node. If multiple nodes are marked, their trees are printed on separate lines.

Without any marks, the pattern "`NP << JJ < /^PP/`" might print this on a match:

```
(NP (NP (DT a) (JJ high) (NN point)) (PP (IN for) (NP (DT the) (NN year))))
```

But "`NP << `JJ < `/^PP/`" would print:

```
(PP (IN for) (NP (DT the) (NN year)))
(JJ high)
```

Note that the subtrees in this example are not printed in the order in which their nodes were specified because the pattern was reordered for greater speed. If it is important that subtrees be printed in a specific order, one can label the nodes and print the labeled nodes using the **-m** option. It is also possible to suppress the reordering of links with the **-d** option.

If there is any negated link between the head and a node, that node cannot be marked for printing. The reason is that, following a successful match, that node will not be aligned with any particular subtree.

If there is a disjunction of links leading to a node, the node may be marked for printing. However, it is possible that the node will not be matched to any subtree. In that case, "`<none>`" is printed.

C.4.2 Formatted output

Although the default output styles should be sufficient for many users, it is also possible to customize the information printed by TGREP2 and the format in which it is printed. The options **-b**, **-s**, and **-m** can be used to set the output formats used at various times.

The **-b** format defines the information printed for each sentence before any matches are attempted. This might be used to print a separator, to print the sentence number, or to print the terminals in the sentence.

The **-s** format defines the information printed for each sentence for which at least one match was successful. Note that this differs from **-b**, which applies to all sentences, regardless of whether there was a match.

The **-m** format is the most useful. It is used every time there is a match between a pattern and a sentence. If the **-f** option was used, there can be at most one match per sentence. Otherwise, there may be more than one match per sentence.

Each of these three formats is a string somewhat like those used by *printf*. The string can contain text, special characters like `\n` or `\t`, and codes representing particular types of fields. Each field code begins with the `%` character. The following fields produce strings or integers:

Non-trees

%f	The name of the corpus file.
%s	The number of the sentence, starting with 1.
%p	The number of the matching pattern, starting with 1.
%i	The number of the current match on the current sentence. The first match to a sentence is 1, the next is 2, and so on.
%j	The number of the current match on the current sentence using the current pattern. This is similar to **%i** but resets every time a new pattern is used.
%c	This prints the comment associated with the current sentence. If there is no comment, nothing will be printed.

If a positive integer N is placed immediately after the `%`, the above fields are right-justified in a column of width N. If the integer is negative, the field is left-justified.

Trees

%h	The head node of the pattern.
%m	The nodes marked for printing. If there are none, the head node is printed. If more than one node is marked for printing, a line break appears between their trees.
%w	The top of the sentence tree.
%=foo=	The node labeled `foo`. Note the = both before and after `foo`.
%N**b**	Prints the sentence tree that is N before the current sentence. N should be a positive integer.
%N**a**	Prints the sentence tree that is N after the current sentence. N should be a positive integer.

Tree printing styles

By default, trees are printed in short form. One of the following characters can be placed immediately after the `%` to print the tree in a different format.

l	Prints the tree in long, multi-line, form.
t	Prints just the terminal symbols.

u Prints just the top (*u*ppermost?) symbol in the tree.

n Prints the node number of the top of the tree in depth-first search order.

x Prints the subtree code, which consists of the sentence number followed by a colon and the node number.

Other than the above fields, any other text in the format is reproduced in the output. Special character codes like `\n` and `\t` can be used to produce newlines or tabs. See the *printf* man page for a complete list of these codes. To produce a single `%` character, `%%` must be specified. No spaces or newlines are printed after fields or trees automatically. Thus, any spaces or line breaks must be specified by the user.

The following example prints a short line, the file name, and sentence number for each sentence for which there is at least one match, followed by the terminals of the previous sentence. For each match, it prints the subtree code for the node labeled `np` followed by a space and the head of the pattern in short form.

```
% tgrep2 -s '----%f %s\n%t1b\n' -m '%x=np= %h\n' 'VP < NP=np'
```

C.4.3 Extracting subtrees using codes

It is often the case that one needs to compare the subtrees matched by a number of minimally different patterns to detect overlap or subtrees matched by one pattern and not by another. For example, one set of patterns might classify relative clauses based on the position of the modified word. Another set of patterns might classify the clauses based on the internal structure of the clause. By intersecting the set of subtree codes for subject-modifying clauses with the set of codes for object-extracted relative clauses, one could obtain the subject-modifying object-extracted clauses.

Rather than storing and comparing the entire subtrees, which may not be unique, it is more efficient to store and compare the subtree codes. Subtree codes are unique identifiers of subtrees in the corpus and are produced when the **-x** option is used or a tree is printed using a formatting field with the **x** prefix.

If the intersection of two sets of subtree codes has been generated, it might be necessary to retrieve the actual clauses corresponding to those codes. That can be done with the **-e** option. When this is used, the name of a file containing subtree codes is given in place of a search pattern. A match occurs on each of those subtrees and the tree can be printed out as if the head of the search pattern matched the tree. Note that the file of subtree codes must be sorted in non-decreasing sentence order. If a "**–**" is given in place of the file name, the codes will be read from the standard input.

C.5 Differences from TGREP

This section reviews the large and small differences between TGREP and TGREP2.

C.5.1 Differences from and unsupported features of TGREP

- There is no distinction between vocab files and corpus files in TGREP2. There are just corpus files.
- TGREP2 does not read commands from a script file, only from the command line.
- The immediate precedence operator does not have the same meaning in TGREP2 as it did in TGREP. This is explained in Section C.3.3.
- In TGREP2, "`(S < NP) << VP`" is not valid syntax for expressing "`S < NP << VP`", as it was in TGREP.
- If no nodes are marked for printing, TGREP2 will print only the subtree matched by the head (first) node in the pattern. TGREP attempted to print all nodes whose subtrees were not descendants of the head node's tree, but did not do it correctly and was thus rather confusing.
- TGREP2 does not have an option to allow a node to be its own sister.
- There is no separate *tprep2* program. The **-p** option is used to generate TGREP2 corpora.

- The **-f**, **-l**, **-c**, **-T**, **-p**, **-P**, **-n**, **-N**, **-s**, and **-e** options to TGREP are subsumed by the use of formatted output in TGREP2.
- The **-v**, **-u**, **-g**, **-O**, **-i**, **-$**, and **-o** options are unsupported in TGREP2.
- TGREP2 uses different flags to represent some of the options also provided by TGREP.

C.5.2 Additional TGREP2 features

- The `,` `,,` `$,` `$,,` `>>,` `>>'` `>`N `>-`N and `>-` links have been added.
- There is now direct support for the `<,` `<'` `>,` and `>'` links.
- `*` can be used instead of `__` to match any node.
- Boolean expressions of relationships to other nodes are allowed.
- Nodes can be labeled, allowing patterns whose link structure is not in tree form and the selection of nodes for printing by label.
- Patterns can be read from files rather than listed on the command line.
- Patterns can be segmented for greater readability and to aid the automatic composition of patterns.
- Multiple patterns can be searched in a single invocation of TGREP2.
- TGREP2 corpus files are up to 20 times smaller than TGREP files.
- Greater control is provided over the printing of context sentences.
- Formatted output allows more information to be printed in a customizable way.
- For most searches, TGREP2 seems to be somewhat faster than TGREP. That is not true, however, for searches involving words that do not occur in the corpus, for which TGREP is specially optimized.

TGREP2 is available for download at...

http://www.cs.cmu.edu/~dr/Tgrep2

Appendix D

Details of the Penglish Language

This appendix includes further information about the Penglish language not given in Chapter 5. Section D.1 contains the actual SLG specification of the Penglish grammar. Section D.2 provides tables of the verb- and noun-usage frequencies that were extracted from the Penn Treebank. The constraint functions in the grammar and some of its structural probabilities are based on the frequency counts given in these lexical entries.

D.1 The Penglish SLG grammar

The following is part of the specification of the Penglish language in the modified stochastic context-free grammar recognized by the SLG program. Included are the definitions of major symbols, their transitions and associated probabilities, and the applications of constraints. Also included are the non-terminals used in producing forms of one of the verbs, *ask*. Not shown, however, are the productions for the other verbs or the actual definitions of the constraint functions, except for a few simple ones. Their inclusion would require 80-some additional pages. Appendix B should help explain the syntax of constraint usage in the SLG grammar.

```
# Top-level Sentences:
S:
         CLS_M1 "." (0.85) |
         CLS_M2 CC  CLS_M2 "." (0.09)  |
         CLS_S2_IFF CLS_M2 "." (0.018) |
         CLS_M2 CLS_S2_IFF "." (0.005) |
         CLS_S2_BEC CLS_M2 "." (0.010) |
         CLS_M2 CLS_S2_BEC "." (0.012) |
         CLS_S2_WHL CLS_M2 "." (0.004) |
         CLS_M2 CLS_S2_WHL "." (0.003) |
         CLS_S2_ALT CLS_M2 "." (0.005) |
         CLS_M2 CLS_S2_ALT "." (0.003) |
         {WHL-TENSE, CLS_S2_WHL CLS_M2 VP2 TENSE, CLS_M2 VP2 TENSE} |
         {IFF-TENSE, CLS_S2_IFF CLS_M2 VP2 TENSE, CLS_M2 VP2 TENSE};

# Main and Subordinate Clauses:
CLS_M1:
         NP1 VP1 |
         {VERB-SBJ, VP.* .VP.* VERB_.*, NP1 NP.* NN};
CLS_M2:
         NP2 VP2 |
         {VERB-SBJ, VP2 .VP.* VERB_.*, NP2 NP.* NN};
CLS_M3:
         NP3 VP3 |
         {VERB-SBJ, VP3 .VP.* VERB_.*, NP3 NP.* NN};
CLS_S2_IFF: if       CLS_M2;
CLS_S2_BEC: because  CLS_M2;
CLS_S2_WHL: while    CLS_M2;
CLS_S2_ALT: although CLS_M2;
```

```
CC:      and (0.65) | but (0.35);

# Sentential Complements:
SC1:
         THAT CLS_M2;
SC2:
         THAT CLS_M3;

# Relative Clauses:
NRC1:
         RP IVP2 (0.149) | NO_RP IVP2_R (0.050) |
         RP TVP2 (0.264) | NO_RP TVP2_R (0.077) |
         RP SVP2 (0.013) | NO_RP SVP2_R (0.002) |
         RP OBR2 (0.045) | NO_RP OBR2   (0.297) |
         RP PVP2 (0.030) | NO_RP PVP2_R (0.073);
ORC1:
         RP IVP2 (0.182) | NO_RP IVP2_R (0.047) |
         RP TVP2 (0.219) | NO_RP TVP2_R (0.082) |
         RP SVP2 (0.017) | NO_RP SVP2_R (0.007) |
         RP OBR2 (0.052) | NO_RP OBR2   (0.231) |
         RP PVP2 (0.063) | NO_RP PVP2_R (0.100);
RC2:
         RP IVP2 (0.136) | NO_RP IVP2_R (0.067) |
         RP TVP3 (0.189) | NO_RP TVP3_R (0.081) |
         RP SVP3 (0.007) | NO_RP SVP3_R (0.003) |
         RP OBR3 (0.052) | NO_RP OBR3   (0.316) |
         RP PVP2 (0.044) | NO_RP PVP2_R (0.105);

# Noun Phrases:
NP1:
         NP              (0.863) |
         NP_M NPP1       (0.102) |
         NP_M NRC1       (0.035) |
         {VERB-SBJ,  NRC1 .*VP2.* VERB_.*,  NP_M NN} |
         {VERB-OBJ,  NRC1 OBR2 VERB_.*, NP_M NN} |
         {NOUN-RP,   NP_M NN, NRC1 .*RP} |
         {NOUN-NPP,  NP_M NN, NPP1} |
         {NOUN-TYPE, NP_M NN, NPP1 TYPE. OP. OP.* NO} |
         {NOUN-NLOC, NP_M NN, NPP1 NLOC. OP. OP.* NO} |
         {NOUN-PURP, NP_M NN, NPP1 PURP. OP. OP.* NO} |
         {NOUN-AUTH, NP_M NN, NPP1 AUTH. OP. OP.* NO} |
         {NOUN-POSS, NP_M NN, NPP1 POSS. OP. OP.* NO} |
         {NOUN-COMP, NP_M NN, NPP1 COMP. OP. OP.* NO};
OP1:
         OP              (0.643) |
         OP_M NPP1       (0.259) |
         OP_M ORC1       (0.098) |
         {VERB-SBJ,  ORC1 .*VP2.* VERB_.*,  OP_M NO} |
         {VERB-OBJ,  ORC1 OBR2 VERB_.*, OP_M NO} |
         {NOUN-RP,   OP_M NO, ORC1 .*RP} |
         {NOUN-NPP,  OP_M NO, NPP1} |
         {NOUN-TYPE, OP_M NO, NPP1 TYPE. OP. OP.* NO} |
         {NOUN-NLOC, OP_M NO, NPP1 NLOC. OP. OP.* NO} |
         {NOUN-PURP, OP_M NO, NPP1 PURP. OP. OP.* NO} |
         {NOUN-AUTH, OP_M NO, NPP1 AUTH. OP. OP.* NO} |
         {NOUN-POSS, OP_M NO, NPP1 POSS. OP. OP.* NO} |
         {NOUN-COMP, OP_M NO, NPP1 COMP. OP. OP.* NO};
NP2:
         NP              (0.874) |
         NP_M NPP2       (0.102) |
         NP_M RC2        (0.024) |
         {VERB-SBJ,  RC2 .*VP.* VERB_.*,  NP_M NN} |
         {VERB-OBJ,  RC2 OBR3 VERB_.*, NP_M NN} |
         {NOUN-RP,   NP_M NN, RC2 .*RP} |
         {NOUN-NPP,  NP_M NN, NPP2} |
         {NOUN-TYPE, NP_M NN, NPP2 TYPE. OP. OP.* NO} |
         {NOUN-NLOC, NP_M NN, NPP2 NLOC. OP. OP.* NO} |
         {NOUN-PURP, NP_M NN, NPP2 PURP. OP. OP.* NO} |
```

```
        {NOUN-AUTH, NP_M NN, NPP2 AUTH. OP. OP.* NO} |
        {NOUN-POSS, NP_M NN, NPP2 POSS. OP. OP.* NO} |
        {NOUN-COMP, NP_M NN, NPP2 COMP. OP. OP.* NO};
OP2:
        OP              (0.874) |
        OP_M NPP2       (0.102) |
        OP_M RC2        (0.024) |
        {VERB-SBJ,  RC2 .*VP.* VERB_.*,  OP_M NO} |
        {VERB-OBJ,  RC2 OBR3 VERB_.*, OP_M NO} |
        {NOUN-RP,   OP_M NO, RC2 .*RP} |
        {NOUN-NPP,  OP_M NO, NPP2} |
        {NOUN-TYPE, OP_M NO, NPP2 TYPE. OP. OP.* NO} |
        {NOUN-NLOC, OP_M NO, NPP2 NLOC. OP. OP.* NO} |
        {NOUN-PURP, OP_M NO, NPP2 PURP. OP. OP.* NO} |
        {NOUN-AUTH, OP_M NO, NPP2 AUTH. OP. OP.* NO} |
        {NOUN-POSS, OP_M NO, NPP2 POSS. OP. OP.* NO} |
        {NOUN-COMP, OP_M NO, NPP2 COMP. OP. OP.* NO};
NP3:
        NP;
OP3 | IO3:
        OP;

# Unmodified Subject Noun Phrase:
NP:
        ART_N ADJ_N NN |
        {NOUN-ART, NN, ART_N} |
        {NOUN-ADJ, NN, ADJ_N ADJ};

# Modified Subject Noun Phrase:
NP_M:
        ART_MN ADJ_NM NN |
        {NOUN-ART, NN, ART_MN} |
        {NOUN-ADJ, NN, ADJ_NM ADJ};

# Unmodified Object Noun Phrase:
OP:
        ART_O ADJ_O NO |
        {NOUN-ART, NO, ART_O} |
        {NOUN-ADJ, NO, ADJ_O ADJ};

# Modified Object Noun Phrase:
OP_M:
        ART_MO ADJ_OM NO |
        {NOUN-ART, NO, ART_MO} |
        {NOUN-ADJ, NO, ADJ_OM ADJ};

# Verb Phrases:
TENSE:  PAST | PRESENT | FUTURE;
TNS_PST: PAST;
TNS_PRS: PRESENT;
TNS_FUT: FUTURE;
PAST | PRESENT | FUTURE: "";

VP1:
        IVP1 (0.151) |
        TVP1 (0.499) |
        SVP1 (0.165) |
        PVP1 (0.185);
VP2:
        TENSE IVP2 (0.151) |
        TENSE TVP2 (0.499) |
        TENSE SVP2 (0.165) |
        TENSE PVP2 (0.185) |
        {TENSE, .VP2 VERB_.* .* TNS_.*, TENSE};
VP3:
        IVP2 (0.151) |
        TVP3 (0.499) |
        PVP2 (0.185);
```

```
IVP1:
        VERB_I VPP1 ADVB |
        {VERB-VPP,  VERB_I, VPP1} |
        {VERB-OFPP, VERB_I, VPP1 OFPP. OP. OP.* NO} |
        {VERB-ONPP, VERB_I, VPP1 ONPP. OP. OP.* NO} |
        {VERB-RECP, VERB_I, VPP1 RECP. OP. OP.* NO} |
        {VERB-PURP, VERB_I, VPP1 PURP. OP. OP.* NO} |
        {VERB-INST, VERB_I, VPP1 INST. OP. OP.* NO} |
        {VERB-DEST, VERB_I, VPP1 DEST. OP. OP.* NO} |
        {VERB-LOCA, VERB_I, VPP1 LOCA. OP. OP.* NO} |
        {VERB-ADVB, VERB_I, ADVB} |
        {TNSE-ADVB, VERB_I .* TNS_.*, ADVB};
IVP2:
        VERB_I VPP2 ADVB |
        {VERB-VPP, VERB_I, VPP2} |
        {VERB-OFPP, VERB_I, VPP2 OFPP. OP. OP.* NO} |
        {VERB-ONPP, VERB_I, VPP2 ONPP. OP. OP.* NO} |
        {VERB-RECP, VERB_I, VPP2 RECP. OP. OP.* NO} |
        {VERB-PURP, VERB_I, VPP2 PURP. OP. OP.* NO} |
        {VERB-INST, VERB_I, VPP2 INST. OP. OP.* NO} |
        {VERB-DEST, VERB_I, VPP2 DEST. OP. OP.* NO} |
        {VERB-LOCA, VERB_I, VPP2 LOCA. OP. OP.* NO} |
        {VERB-ADVB, VERB_I, ADVB} |
        {TNSE-ADVB, VERB_I .* TNS_.*, ADVB};
IVP2_R:
        VERB_IR VPP2 ADVB |
        {VERB-VPP, VERB_IR, VPP2} |
        {VERB-OFPP, VERB_IR, VPP2 OFPP. OP. OP.* NO} |
        {VERB-ONPP, VERB_IR, VPP2 ONPP. OP. OP.* NO} |
        {VERB-RECP, VERB_IR, VPP2 RECP. OP. OP.* NO} |
        {VERB-PURP, VERB_IR, VPP2 PURP. OP. OP.* NO} |
        {VERB-INST, VERB_IR, VPP2 INST. OP. OP.* NO} |
        {VERB-DEST, VERB_IR, VPP2 DEST. OP. OP.* NO} |
        {VERB-LOCA, VERB_IR, VPP2 LOCA. OP. OP.* NO} |
        {VERB-ADVB, VERB_IR, ADVB} |
        {TNSE-ADVB, VERB_IR .* TNS_.*, ADVB};

TVP1:
        VERB_T OP1 VPP1 ADVB (0.925) |
        VERB_G OP2 OP1  ADVB (0.075) |
        {VERB-OBJ,  VERB_[TG], OP1 OP.* NO} |
        {VERB-IO,   VERB_G, OP2 OP.* NO} |
        {VERB-VPP,  VERB_[TG], VPP1 } |
        {VERB-OFPP, VERB_[TG], VPP1 OFPP. OP. OP.* NO} |
        {VERB-ONPP, VERB_[TG], VPP1 ONPP. OP. OP.* NO} |
        {VERB-RECP, VERB_[TG], VPP1 RECP. OP. OP.* NO} |
        {VERB-PURP, VERB_[TG], VPP1 PURP. OP. OP.* NO} |
        {VERB-INST, VERB_[TG], VPP1 INST. OP. OP.* NO} |
        {VERB-DEST, VERB_[TG], VPP1 DEST. OP. OP.* NO} |
        {VERB-LOCA, VERB_[TG], VPP1 LOCA. OP. OP.* NO} |
        {VERB-ADVB, VERB_[TG], ADVB} |
        {TNSE-ADVB, VERB_[TG] .* TNS_.*, ADVB};
TVP2:
        VERB_T OP2 VPP2 ADVB (0.925) |
        VERB_G IO3 OP2  ADVB (0.075) |
        {VERB-OBJ,  VERB_[TG], OP2 OP.* NO} |
        {VERB-IO,   VERB_G, IO3 OP.* NO} |
        {VERB-VPP,  VERB_[TG], VPP2} |
        {VERB-OFPP, VERB_[TG], VPP2 OFPP. OP. OP.* NO} |
        {VERB-ONPP, VERB_[TG], VPP2 ONPP. OP. OP.* NO} |
        {VERB-RECP, VERB_[TG], VPP2 RECP. OP. OP.* NO} |
        {VERB-PURP, VERB_[TG], VPP2 PURP. OP. OP.* NO} |
        {VERB-INST, VERB_[TG], VPP2 INST. OP. OP.* NO} |
        {VERB-DEST, VERB_[TG], VPP2 DEST. OP. OP.* NO} |
        {VERB-LOCA, VERB_[TG], VPP2 LOCA. OP. OP.* NO} |
        {VERB-ADVB, VERB_[TG], ADVB} |
        {TNSE-ADVB, VERB_[TG] .* TNS_.*, ADVB};
```

```
TVP3:
         VERB_T OP3 VPP2 ADVB (0.925) |
         VERB_G IO3 OP3  ADVB (0.075) |
         {VERB-OBJ,  VERB_[TG], OP3 OP.* NO} |
         {VERB-IO,   VERB_G, IO3 OP.* NO} |
         {VERB-VPP,  VERB_[TG], VPP2} |
         {VERB-OFPP, VERB_[TG], VPP2 OFPP. OP. OP.* NO} |
         {VERB-ONPP, VERB_[TG], VPP2 ONPP. OP. OP.* NO} |
         {VERB-RECP, VERB_[TG], VPP2 RECP. OP. OP.* NO} |
         {VERB-PURP, VERB_[TG], VPP2 PURP. OP. OP.* NO} |
         {VERB-INST, VERB_[TG], VPP2 INST. OP. OP.* NO} |
         {VERB-DEST, VERB_[TG], VPP2 DEST. OP. OP.* NO} |
         {VERB-LOCA, VERB_[TG], VPP2 LOCA. OP. OP.* NO} |
         {VERB-ADVB, VERB_[TG], ADVB} |
         {TNSE-ADVB, VERB_[TG] .* TNS_.*, ADVB};
TVP2_R:
         VERB_TR OP2 VPP2 ADVB (0.925) |
         VERB_GR IO3 OP2  ADVB (0.075) |
         {VERB-OBJ,  VERB_[TG]R, OP2 OP.* NO} |
         {VERB-IO,   VERB_GR, IO3 OP.* NO} |
         {VERB-VPP,  VERB_[TG]R, VPP2} |
         {VERB-OFPP, VERB_[TG]R, VPP2 OFPP. OP. OP.* NO} |
         {VERB-ONPP, VERB_[TG]R, VPP2 ONPP. OP. OP.* NO} |
         {VERB-RECP, VERB_[TG]R, VPP2 RECP. OP. OP.* NO} |
         {VERB-PURP, VERB_[TG]R, VPP2 PURP. OP. OP.* NO} |
         {VERB-INST, VERB_[TG]R, VPP2 INST. OP. OP.* NO} |
         {VERB-DEST, VERB_[TG]R, VPP2 DEST. OP. OP.* NO} |
         {VERB-LOCA, VERB_[TG]R, VPP2 LOCA. OP. OP.* NO} |
         {VERB-ADVB, VERB_[TG]R, ADVB} |
         {TNSE-ADVB, VERB_[TG]R .* TNS_.*, ADVB};
TVP3_R:
         VERB_TR OP3 VPP2 ADVB (0.925) |
         VERB_GR IO3 OP3  ADVB (0.075) |
         {VERB-OBJ,  VERB_[TG]R, OP3 OP.* NO} |
         {VERB-IO,   VERB_GR, IO3 OP.* NO} |
         {VERB-VPP,  VERB_[TG]R, VPP2} |
         {VERB-OFPP, VERB_[TG]R, VPP2 OFPP. OP. OP.* NO} |
         {VERB-ONPP, VERB_[TG]R, VPP2 ONPP. OP. OP.* NO} |
         {VERB-RECP, VERB_[TG]R, VPP2 RECP. OP. OP.* NO} |
         {VERB-PURP, VERB_[TG]R, VPP2 PURP. OP. OP.* NO} |
         {VERB-INST, VERB_[TG]R, VPP2 INST. OP. OP.* NO} |
         {VERB-DEST, VERB_[TG]R, VPP2 DEST. OP. OP.* NO} |
         {VERB-LOCA, VERB_[TG]R, VPP2 LOCA. OP. OP.* NO} |
         {VERB-ADVB, VERB_[TG]R, ADVB} |
         {TNSE-ADVB, VERB_[TG]R .* TNS_.*, ADVB};

SVP1:
         VERB_S SC1 ADVB (0.337) |
         VERB_ST IO3 SC1 ADVB (0.051) |
         {VERB-THAT, VERB_S.*, SC1 THAT |
         {VERB-IO,   VERB_ST, IO3 OP.* NO} |
         {VERB-ADVB, VERB_S.*, ADVB} |
         {TNSE-ADVB, VERB_S.* .* TNS_.*, ADVB};
SVP2:
         VERB_S SC2 ADVB (0.337) |
         VERB_ST IO3 SC2 ADVB (0.051) |
         {VERB-THAT, VERB_S.*, SC2 THAT} |
         {VERB-IO,   VERB_ST, IO3 OP.* NO} |
         {VERB-ADVB, VERB_S.*, ADVB} |
         {TNSE-ADVB, VERB_S.* .* TNS_.*, ADVB};
SVP2_R:
         VERB_SR SC2 ADVB (0.580) |
         VERB_STR IO3 SC2 ADVB (0.031) |
         {VERB-THAT, VERB_S.*, SC2 THAT} |
         {VERB-IO,   VERB_STR, IO3 OP.* NO} |
         {VERB-ADVB, VERB_S.*, ADVB} |
         {TNSE-ADVB, VERB_S.* .* TNS_.*, ADVB};
SVP3:
```

```
        VERB_S SC2 ADVB (0.337) |
        VERB_ST IO3 SC2 ADVB (0.051) |
        {VERB-THAT, VERB_S.*, SC2 THAT} |
        {VERB-IO,   VERB_ST, IO3 OP.* NO} |
        {VERB-ADVB, VERB_S.*, ADVB} |
        {TNSE-ADVB, VERB_S.* .* TNS_.*, ADVB};
SVP3_R:
        VERB_SR SC2 ADVB (0.580) |
        VERB_STR IO3 SC2 ADVB (0.031) |
        {VERB-THAT, VERB_S.*, SC2 THAT} |
        {VERB-IO,   VERB_STR, IO3 OP.* NO} |
        {VERB-ADVB, VERB_S.*, ADVB} |
        {TNSE-ADVB, VERB_S.* .* TNS_.*, ADVB};

OBR2:
        NP2 VERB_T VPP2 ADVB (0.925) |
        NP2 VERB_G IO3  ADVB (0.075) |
        {VERB-SBJ,  VERB_[TG], NP2 NP.* NN} |
        {VERB-IO,   VERB_G, IO3 OP.* NO} |
        {VERB-VPP,  VERB_[TG], VPP2} |
        {VERB-OFPP, VERB_[TG], VPP2 OFPP. OP. OP.* NO} |
        {VERB-ONPP, VERB_[TG], VPP2 ONPP. OP. OP.* NO} |
        {VERB-RECP, VERB_[TG], VPP2 RECP. OP. OP.* NO} |
        {VERB-PURP, VERB_[TG], VPP2 PURP. OP. OP.* NO} |
        {VERB-INST, VERB_[TG], VPP2 INST. OP. OP.* NO} |
        {VERB-DEST, VERB_[TG], VPP2 DEST. OP. OP.* NO} |
        {VERB-LOCA, VERB_[TG], VPP2 LOCA. OP. OP.* NO} |
        {VERB-ADVB, VERB_[TG], ADVB} |
        {TNSE-ADVB, VERB_[TG] .* TNS_.*, ADVB};
OBR3:
        NP3 VERB_T VPP2 ADVB (0.925) |
        NP3 VERB_G IO3  ADVB (0.075) |
        {VERB-SBJ,  VERB_[TG], NP3 NP.* NN} |
        {VERB-IO,   VERB_G, IO3 OP.* NO} |
        {VERB-VPP,  VERB_[TG], VPP2} |
        {VERB-OFPP, VERB_[TG], VPP2 OFPP. OP. OP.* NO} |
        {VERB-ONPP, VERB_[TG], VPP2 ONPP. OP. OP.* NO} |
        {VERB-RECP, VERB_[TG], VPP2 RECP. OP. OP.* NO} |
        {VERB-PURP, VERB_[TG], VPP2 PURP. OP. OP.* NO} |
        {VERB-INST, VERB_[TG], VPP2 INST. OP. OP.* NO} |
        {VERB-DEST, VERB_[TG], VPP2 DEST. OP. OP.* NO} |
        {VERB-LOCA, VERB_[TG], VPP2 LOCA. OP. OP.* NO} |
        {VERB-ADVB, VERB_[TG], ADVB} |
        {TNSE-ADVB, VERB_[TG] .* TNS_.*, ADVB};

PVP1:
        VERB_P VPP1 ADVB (0.912) |
        VERB_P by OP2 VPP1 ADVB (0.088) |
        {VERB-OBJ,  VERB_P, OP2 OP.* NO} |
        {VERB-VPP,  VERB_P, VPP1} |
        {VERB-OFPP, VERB_P, VPP1 OFPP. OP. OP.* NO} |
        {VERB-ONPP, VERB_P, VPP1 ONPP. OP. OP.* NO} |
        {VERB-RECP, VERB_P, VPP1 RECP. OP. OP.* NO} |
        {VERB-PURP, VERB_P, VPP1 PURP. OP. OP.* NO} |
        {VERB-INST, VERB_P, VPP1 INST. OP. OP.* NO} |
        {VERB-DEST, VERB_P, VPP1 DEST. OP. OP.* NO} |
        {VERB-LOCA, VERB_P, VPP1 LOCA. OP. OP.* NO} |
        {VERB-ADVB, VERB_P, ADVB} |
        {TNSE-ADVB, VERB_P .* TNS_.*, ADVB};
PVP2:
        VERB_P VPP2 ADVB (0.912) |
        VERB_P by OP3 VPP2 ADVB (0.088) |
        {VERB-OBJ,  VERB_P, OP3 OP.* NO} |
        {VERB-VPP,  VERB_P, VPP2} |
        {VERB-OFPP, VERB_P, VPP2 OFPP. OP. OP.* NO} |
        {VERB-ONPP, VERB_P, VPP2 ONPP. OP. OP.* NO} |
        {VERB-RECP, VERB_P, VPP2 RECP. OP. OP.* NO} |
        {VERB-PURP, VERB_P, VPP2 PURP. OP. OP.* NO} |
```

```
        {VERB-INST, VERB_P, VPP2 INST. OP. OP.* NO} |
        {VERB-DEST, VERB_P, VPP2 DEST. OP. OP.* NO} |
        {VERB-LOCA, VERB_P, VPP2 LOCA. OP. OP.* NO} |
        {VERB-ADVB, VERB_P, ADVB} |
        {TNSE-ADVB, VERB_P .* TNS_.*, ADVB};
PVP2_R:
        VERB_PR VPP2 ADVB (0.789) |
        VERB_PR by OP3 VPP2 ADVB (0.211) |
        {VERB-OBJ,  VERB_PR, OP3 OP.* NO} |
        {VERB-VPP,  VERB_PR, VPP2} |
        {VERB-OFPP, VERB_PR, VPP2 OFPP. OP. OP.* NO} |
        {VERB-ONPP, VERB_PR, VPP2 ONPP. OP. OP.* NO} |
        {VERB-RECP, VERB_PR, VPP2 RECP. OP. OP.* NO} |
        {VERB-PURP, VERB_PR, VPP2 PURP. OP. OP.* NO} |
        {VERB-INST, VERB_PR, VPP2 INST. OP. OP.* NO} |
        {VERB-DEST, VERB_PR, VPP2 DEST. OP. OP.* NO} |
        {VERB-LOCA, VERB_PR, VPP2 LOCA. OP. OP.* NO} |
        {VERB-ADVB, VERB_PR, ADVB} |
        {TNSE-ADVB, VERB_PR .* TNS_.*, ADVB};

# Verb-modifying Prepositional Phrases:
VPP1:
        "" | OFPP1 | RECP1 | PURP1 | INST1 | DEST1 | LOCA1 | ONPP1;
VPP2:
        "" | OFPP2 | RECP1 | PURP2 | INST2 | DEST2 | LOCA2 | ONPP2;

OFPP1:  of OP2;
OFPP2:  of OP3;

ONPP1:  on OP2;
ONPP2:  on OP3;

RECP1:  to OP2;
RECP2:  to OP3;

PURP1:  for OP2;
PURP2:  for OP3;

INST1:  with OP2;
INST2:  with OP3;

DEST1:  to OP2;
DEST2:  to OP3;

LOCA1:  in OP2;
LOCA2:  in OP3;

ADVB:
        "" | yesterday | tomorrow | quickly | fiercely | loudly;

# Noun-modifying Prepositional Phrases:
NPP1:
        TYPE1 | PURP1 | AUTH1 | POSS1 | COMP1 | NLOC1;
NPP2:
        TYPE2 | PURP2 | AUTH2 | POSS2 | COMP2 | NLOC2;

TYPE1:  of OP2;
TYPE2:  of OP3;

AUTH1:  by OP2;
AUTH2:  by OP3;

POSS1:  with OP2;
POSS2:  with OP3;

COMP1:  with OP2;
COMP2:  with OP3;
```

```
NLOC1:   on OP2;
NLOC2:   on OP3;

# Closed Class:
THAT:
         "" | that | if;
RP:
         that (0.700) |
         who | which;
NO_RP:
         NONE;
NONE:
         "";

ART_N:
         ART_NS  | ART_NP | ART_NM | "";
ART_NS:
         the   (0.794) |
         this  (0.045) |
         that  (0.018) |
         a     (0.143);
ART_NP:
         ""    (0.720) |
         the   (0.210) |
         these (0.029) |
         those (0.008) |
         some  (0.033);
ART_NM:
         ""    (0.720) |
         the   (0.215) |
         this  (0.040) |
         that  (0.025);

ART_MN:
         ART_MNS | ART_MNP | ART_MNM | "";
ART_MNS:
         the   (0.709) |
         this  (0.024) |
         that  (0.006) |
         a     (0.261);
ART_MNP:
         ""    (0.691) |
         the   (0.277) |
         these (0.009) |
         those (0.006) |
         some  (0.017);
ART_MNM:
         ""    (0.700) |
         the   (0.250) |
         this  (0.030) |
         that  (0.020);

ART_O:
         ART_OS  | ART_OP | ART_OM | "";
ART_OS:
         the   (0.423) |
         this  (0.045) |
         that  (0.014) |
         a     (0.518);
ART_OP:
         ""    (0.822) |
         the   (0.142) |
         these (0.014) |
         those (0.006) |
         some  (0.016);
ART_OM:
         ""    (0.820) |
         the   (0.144) |
```

```
         this  (0.024) |
         that  (0.012);

ART_MO:
         ART_MOS | ART_MOP | ART_MOM | "";
ART_MOS:
         the   (0.397) |
         this  (0.009) |
         that  (0.003) |
         a     (0.591);
ART_MOP:
         ""    (0.777) |
         the   (0.199) |
         these (0.003) |
         those (0.004) |
         some  (0.017);
ART_MOM:
         ""    (0.780) |
         the   (0.204) |
         this  (0.007) |
         that  (0.009);

# Adjectives:
ADJ_N:
         "" | ADJ (0.010);
ADJ_NM:
         "" | ADJ (0.235);
ADJ_O:
         "" | ADJ (0.233);
ADJ_OM:
         "" | ADJ (0.325);
ADJ:
         nice_K | nice_N | mean | fierce | small | big | loud | hard_D |
         hard_T | new | old | young;

# Base Functions:

TENSE {
  PAST    : PAST;
  PRESENT : PRESENT;
  FUTURE  : FUTURE;
}

WHL-TENSE {
  PAST    : PAST;
  PRESENT : PRESENT | FUTURE;
  FUTURE  : FUTURE;
}

IFF-TENSE {
  PAST    : FUTURE;
  PRESENT : PRESENT | FUTURE;
  FUTURE  : FUTURE;
}

TNSE-ADVB {
  PAST    ! tomorrow;
  PRESENT ! yesterday | tomorrow;
  FUTURE  ! yesterday;

######## Verb Info ########

VERB_I:
        ASK_P_AS (0.0431121) | ASK_P_AP (0.0261076) | ASK_P_AI (0.00678824) |
        BELIEVE_AS (0.0153693) | BELIEVE_AP (0.00930722) | BELIEVE_AI (0.00241998) |
        BARK_AS (0.0065124) | BARK_AP (0.00394374) | BITE_AS (0.00520992) |
        BITE_AP (0.00315499) | BUY_AS (0.0121131) | BUY_AP (0.00733536) |
        BUY_AI (0.00190727) | CONSIST_AS (0.0208397) | CONSIST_AP (0.01262) |
```

```
        DRIVE_H_AS (0.00781488) | DRIVE_H_AP (0.00473249) | DRIVE_H_AI (0.0012305) |
        DRIVE_C_AS (0.00429818) | DRIVE_C_AP (0.00260287) | EAT_AS (0.00833587) |
        EAT_AP (0.00504799) | EAT_AI (0.00131253) | FISH_AS (0.00390744) |
        FISH_AP (0.00236624) | FISH_AI (0.000615248) | FOLLOW_AS (0.0216212) |
        FOLLOW_AP (0.0130932) | FOLLOW_AI (0.00340437) | FORGET_M_AS (0.00403769) |
        FORGET_M_AP (0.00244512) | FORGET_M_AI (0.000635756) | FLY_H_AS (0.00390744) |
        FLY_H_AP (0.00236624) | FLY_H_AI (0.000615248) | FLY_P_AS (0.0105501) |
        FLY_P_AP (0.00638886) | GIVE_AS (0.00846612) | GIVE_AP (0.00512686) |
        GIVE_AI (0.00133304) | GO_AS (0.130248) | GO_AP (0.0788748) |
        GO_AI (0.0205083) | GUESS_AS (0.0032562) | GUESS_AP (0.00197187) |
        GUESS_AI (0.000512707) | HEAR_AS (0.0122433) | HEAR_AP (0.00741423) |
        HEAR_AI (0.00192778) | HIT_H_AS (0.00286546) | HIT_H_AP (0.00173525) |
        HIT_H_AI (0.000451182) | HIT_O_AS (0.00130248) | HIT_O_AP (0.000788748) |
        HOPE_AS (0.0242261) | HOPE_AP (0.0146707) | HOPE_AI (0.00381454) |
        KILL_H_AS (0.00260496) | KILL_H_AP (0.0015775) | KILL_H_AI (0.000410165) |
        KILL_A_AS (0.00260496) | KILL_A_AP (0.0015775) | KNOW_O_AS (0.00520992) |
        KNOW_O_AP (0.00315499) | KNOW_O_AI (0.000820331) | KNOW_T_AS (0.0390744) |
        KNOW_T_AP (0.0236624) | KNOW_T_AI (0.00615248) | LEAVE_I_AS (0.0278731) |
        LEAVE_I_AP (0.0168792) | LEAVE_I_AI (0.00438877) | PARK_AS (0.00065124) |
        PARK_AP (0.000394374) | PARK_AI (0.000102541) | PLAY_I_AS (0.0192767) |
        PLAY_I_AP (0.0116735) | PLAY_I_AI (0.00303522) | READ_AS (0.00898711) |
        READ_AP (0.00544236) | READ_AI (0.00141507) | SEE_AS (0.0398559) |
        SEE_AP (0.0241357) | SEE_AI (0.00627553) | TELL_S_AS (0.0065124) |
        TELL_S_AP (0.00394374) | TELL_S_AI (0.00102541) | THINK_AS (0.0669475) |
        THINK_AP (0.0405416) | THINK_AI (0.0105412) | THROW_AS (0.00195372) |
        THROW_AP (0.00118312) | THROW_AI (0.000307624);

VERB_IR:
        ASK_P_AR (0.0753986) | BELIEVE_AR (0.0268793) | BARK_AR (0.0113895) |
        BITE_AR (0.00911162) | BUY_AR (0.0211845) | CONSIST_AR (0.0364465) |
        DRIVE_H_AR (0.0136674) | DRIVE_C_AR (0.00751708) | EAT_AR (0.0145786) |
        FISH_AR (0.00683371) | FOLLOW_AR (0.0378132) | FORGET_M_AR (0.0070615) |
        FLY_H_AR (0.00683371) | FLY_P_AR (0.018451) | GIVE_AR (0.0148064) |
        GO_AR (0.22779) | GUESS_AR (0.00569476) | HEAR_AR (0.0214123) |
        HIT_H_AR (0.00501139) | HIT_O_AR (0.0022779) | HOPE_AR (0.042369) |
        KILL_H_AR (0.00455581) | KILL_A_AR (0.00455581) |
        KNOW_O_AR (0.00911162) | KNOW_T_AR (0.0683371) | LEAVE_I_AR (0.0487472) |
        PARK_AR (0.00113895) | PLAY_I_AR (0.033713) | READ_AR (0.0157175) |
        SEE_AR (0.0697039) | TELL_S_AR (0.0113895) | THINK_AR (0.117084) |
        THROW_AR (0.00341686);

VERB_T:
        ASK_P_AS (0.00872476) | ASK_P_AP (0.00528349) | ASK_P_AI (0.00137376) |
        ASK_Q_AS (0.00501674) | ASK_Q_AP (0.00303801) | ASK_Q_AI (0.000789913) |
        BELIEVE_AS (0.00209394) | BELIEVE_AP (0.00126804) | BELIEVE_AI (0.000329703) |
        BITE_AS (0.00218119) | BITE_AP (0.00132087) | BUY_AS (0.0394359) |
        BUY_AP (0.0238814) | BUY_AI (0.0062094) | DRIVE_H_AS (0.00488587) |
        DRIVE_H_AP (0.00295875) | DRIVE_H_AI (0.000769307) | DRIVE_C_AS (0.000436238) |
        DRIVE_C_AP (0.000264174) | EAT_AS (0.00296642) | EAT_AP (0.00179639) |
        EAT_AI (0.000467079) | EXAMINE_AS (0.00292279) | EXAMINE_AP (0.00176997) |
        EXAMINE_AI (0.00046021) | FEEL_AS (0.00667444) | FEEL_AP (0.00404187) |
        FEEL_AI (0.00105093) | FIND_AS (0.0314964) | FIND_AP (0.0190734) |
        FIND_AI (0.00495928) | FOLLOW_AS (0.0155301) | FOLLOW_AP (0.00940461) |
        FOLLOW_AI (0.0024453) | FORGET_M_AS (0.0009161) | FORGET_M_AP (0.000554766) |
        FORGET_M_AI (0.000144245) | FORGET_L_AS (0.000872476) | FORGET_L_AP (0.000528349) |
        FORGET_L_AI (0.000137376) | FLY_H_AS (0.0009161) | FLY_H_AP (0.000554766) |
        FLY_H_AI (0.000144245) | GET_AS (0.0436238) | GET_AP (0.0264174) |
        GET_AI (0.00686881) | GIVE_AS (0.0191945) | GIVE_AP (0.0116237) |
        GIVE_AI (0.00302228) | GUESS_AS (0.00034899) | GUESS_AP (0.00021134) |
        GUESS_AI (5.49505e-05) | HAVE_AS (0.0436238) | HAVE_AP (0.0264174) |
        HAVE_AI (0.00686881) | HEAR_AS (0.00985898) | HEAR_AP (0.00597034) |
        HEAR_AI (0.00155235) | HIT_H_AS (0.00436238) | HIT_H_AP (0.00264174) |
        HIT_H_AI (0.000686881) | HIT_O_AS (0.00353353) | HIT_O_AP (0.00213981) |
        INVOLVE_AS (0.00998985) | INVOLVE_AP (0.00604959) | KILL_H_AS (0.00671807) |
        KILL_H_AP (0.00406829) | KILL_H_AI (0.0010578) | KILL_A_AS (0.00671807) |
        KILL_A_AP (0.00406829) | KNOW_P_AS (0.0130871) | KNOW_P_AP (0.00792523) |
        KNOW_P_AI (0.00206064) | KNOW_T_AS (0.00680531) | KNOW_T_AP (0.00412112) |
        KNOW_T_AI (0.00107153) | LEAVE_I_AS (0.00872476) | LEAVE_I_AP (0.00528349) |
```

```
        LEAVE_I_AI (0.00137376) | LEAVE_T_AS (0.0117784) | LEAVE_T_AP (0.00713271) |
        LEAVE_T_AI (0.00185458) | PARK_AS (0.000305367) | PARK_AP (0.000184922) |
        PARK_AI (4.80817e-05) | PLAY_T_AS (0.0130435) | PLAY_T_AP (0.00789881) |
        PLAY_T_AI (0.00205377) | PUT_AS (0.0292279) | PUT_AP (0.0176997) |
        PUT_AI (0.0046021) | QUESTION_AS (0.00244293) | QUESTION_AP (0.00147938) |
        QUESTION_AI (0.000384653) | READ_AS (0.00841939) | READ_AP (0.00509857) |
        READ_AI (0.00132568) | SAY_AS (0.00222481) | SAY_AP (0.00134729) |
        SAY_AI (0.000350309) | SEE_AS (0.0453688) | SEE_AP (0.0274741) |
        SEE_AI (0.00714356) | SHOW_AS (0.0201542) | SHOW_AP (0.0122049) |
        SHOW_AI (0.00317339) | TAKE_AS (0.0436238) | TAKE_AP (0.0264174) |
        TAKE_AI (0.00686881) | TELL_S_AS (0.0130871) | TELL_S_AP (0.00792523) |
        TELL_S_AI (0.00206064) | TELL_P_AS (0.00410064) | TELL_P_AP (0.00248324) |
        TELL_P_AI (0.000645668) | THINK_AS (0.00270468) | THINK_AP (0.00163788) |
        THINK_AI (0.000425866) | THROW_AS (0.00567109) | THROW_AP (0.00343427) |
        THROW_AI (0.000892945) | USE_AS (0.0412681) | USE_AP (0.0249909) |
        USE_AI (0.00649789) | WANT_AS (0.0141341) | WANT_AP (0.00855925) |
        WANT_AI (0.00222549) | WRITE_S_AS (0.00567109) | WRITE_S_AP (0.00343427) |
        WRITE_S_AI (0.000892945) | WRITE_P_AS (0.00436238) | WRITE_P_AP (0.00264174) |
        WRITE_P_AI (0.000686881);

VERB_TR:
        ASK_P_AR (0.0153268) | ASK_Q_AR (0.00881294) | BELIEVE_AR (0.00367844) |
        BITE_AR (0.00383171) | BUY_AR (0.0692773) | DRIVE_H_AR (0.00858303) |
        DRIVE_C_AR (0.000766342) | EAT_AR (0.00521113) | EXAMINE_AR (0.00513449) |
        FEEL_AR (0.011725) | FIND_AR (0.0553299) | FOLLOW_AR (0.0272818) |
        FORGET_M_AR (0.00160932) | FORGET_L_AR (0.00153268) | FLY_H_AR (0.00160932) |
        GET_AR (0.0766342) | GIVE_AR (0.0337191) | GUESS_AR (0.000613074) |
        HAVE_AR (0.0766342) | HEAR_AR (0.0173193) | HIT_H_AR (0.00766342) |
        HIT_O_AR (0.00620737) | INVOLVE_AR (0.0175492) | KILL_H_AR (0.0118017) |
        KILL_A_AR (0.0118017) | KNOW_P_AR (0.0229903) | KNOW_T_AR (0.0119549) |
        LEAVE_I_AR (0.0153268) | LEAVE_T_AR (0.0206912) | PARK_AR (0.00053644) |
        PLAY_T_AR (0.0229136) | PUT_AR (0.0513449) | QUESTION_AR (0.00429152) |
        READ_AR (0.0147904) | SAY_AR (0.00390835) | SEE_AR (0.0796996) |
        SHOW_AR (0.035405) | TAKE_AR (0.0766342) | TELL_S_AR (0.0229903) |
        TELL_P_AR (0.00720362) | THINK_AR (0.00475132) | THROW_AR (0.00996245) |
        USE_AR (0.072496) | WANT_AR (0.0248295) | WRITE_S_AR (0.00996245) |
        WRITE_P_AR (0.00766342);

VERB_G:
        BUY_AS (0.0038638) | BUY_AP (0.00233982) | BUY_AI (0.000608376) |
        GET_AS (0.077276) | GET_AP (0.0467963) | GET_AI (0.0121675) |
        GIVE_AS (0.348515) | GIVE_AP (0.211051) | GIVE_AI (0.0548755) |
        READ_AS (0.0231828) | READ_AP (0.0140389) | READ_AI (0.00365026) |
        SAY_AS (0.0038638) | SAY_AP (0.00233982) | SAY_AI (0.000608376) |
        SHOW_AS (0.0119778) | SHOW_AP (0.00725343) | SHOW_AI (0.00188597) |
        TAKE_AS (0.046752) | TAKE_AP (0.0283118) | TAKE_AI (0.00736135) |
        TELL_S_AS (0.0367061) | TELL_S_AP (0.0222282) | TELL_S_AI (0.00577957) |
        THROW_AS (0.00695484) | THROW_AP (0.00421167) | THROW_AI (0.00109508) |
        WRITE_S_AS (0.00811398) | WRITE_S_AP (0.00491361) | WRITE_S_AI (0.00127759);

VERB_GR:
        BUY_AR (0.00681199) | GET_AR (0.13624) | GIVE_AR (0.614441) |
        READ_AR (0.0408719) | SAY_AR (0.00681199) | SHOW_AR (0.0211172) |
        TAKE_AR (0.0824251) | TELL_S_AR (0.0647139) | THROW_AR (0.0122616) |
        WRITE_S_AR (0.0143052);

VERB_S:
        ASK_Q_AS (0.0150091) | ASK_Q_AP (0.00908914) | ASK_Q_AI (0.00236327) |
        BELIEVE_AS (0.0642251) | BELIEVE_AP (0.0388931) | BELIEVE_AI (0.0101126) |
        FEEL_AS (0.0321126) | FEEL_AP (0.0194465) | FEEL_AI (0.0050563) |
        FIND_AS (0.049565) | FIND_AP (0.0300153) | FIND_AI (0.00780429) |
        FORGET_M_AS (0.00523574) | FORGET_M_AP (0.00317063) | FORGET_M_AI (0.000824397) |
        FORGET_L_AS (0.00523574) | FORGET_L_AP (0.00317063) | FORGET_L_AI (0.000824397) |
        GUESS_AS (0.0034905) | GUESS_AP (0.00211375) | GUESS_AI (0.000549598) |
        HEAR_AS (0.00523574) | HEAR_AP (0.00317063) | HEAR_AI (0.000824397) |
        HOPE_AS (0.0195468) | HOPE_AP (0.011837) | HOPE_AI (0.00307775) |
        KNOW_T_AS (0.101922) | KNOW_T_AP (0.0617216) | KNOW_T_AI (0.0160483) |
        QUESTION_AS (0.00733004) | QUESTION_AP (0.00443888) | QUESTION_AI (0.00115416) |
```

```
        READ_AS (0.0027924) | READ_AP (0.001691) | READ_AI (0.000439678) |
        REALIZE_AS (0.028273) | REALIZE_AP (0.0171214) | REALIZE_AI (0.00445174) |
        SAY_AS (0.0527065) | SAY_AP (0.0319177) | SAY_AI (0.00829893) |
        SEE_AS (0.0544517) | SEE_AP (0.0329745) | SEE_AI (0.00857373) |
        SHOW_AS (0.0628289) | SHOW_AP (0.0380476) | SHOW_AI (0.00989276) |
        TELL_S_AS (0.00663194) | TELL_S_AP (0.00401613) | TELL_S_AI (0.00104424) |
        THINK_AS (0.0380464) | THINK_AP (0.0230399) | THINK_AI (0.00599062) |
        WISH_AS (0.0020943) | WISH_AP (0.00126825) | WISH_AI (0.000329759) |
        WRITE_S_AS (0.0104715) | WRITE_S_AP (0.00634126) | WRITE_S_AI (0.00164879);

VERB_SR:
        BELIEVE_AR (0.102399) | FEEL_AR (0.037952) | FIND_AR (0.0200501) |
        FORGET_M_AR (0.0028643) | FORGET_L_AR (0.0028643) | GUESS_AR (0.0157537) |
        HEAR_AR (0.0028643) | HOPE_AR (0.0318654) | KNOW_T_AR (0.0769782) |
        READ_AR (0.00250627) | REALIZE_AR (0.0161117) | SAY_AR (0.371285) |
        SEE_AR (0.0075188) | SHOW_AR (0.0168278) | TELL_S_AR (0.00214823) |
        THINK_AR (0.264948) | WISH_AR (0.0218403) | WRITE_S_AR (0.00322234);

VERB_ST:
        ASK_Q_AS (0.0941515) | ASK_Q_AP (0.0570157) | ASK_Q_AI (0.0148247) |
        BELIEVE_AS (0.00229638) | BELIEVE_AP (0.00139063) | BELIEVE_AI (0.000361577) |
        FIND_AS (0.00918551) | FIND_AP (0.00556251) | FIND_AI (0.00144631) |
        READ_AS (0.00229638) | READ_AP (0.00139063) | READ_AI (0.000361577) |
        SHOW_AS (0.0206674) | SHOW_AP (0.0125156) | SHOW_AI (0.0032542) |
        TELL_S_AS (0.42483) | TELL_S_AP (0.257266) | TELL_S_AI (0.0668918) |
        WRITE_S_AS (0.0137783) | WRITE_S_AP (0.00834376) | WRITE_S_AI (0.00216946);

VERB_STR:
        BELIEVE_AR (0.0133333) | FIND_AR (0.00666667) | SHOW_AR (0.0133333) |
        TELL_S_AR (0.953333) | WRITE_S_AR (0.0133333);

VERB_P:
        ASK_P_PS (0.0114862) | ASK_P_PP (0.00695573) | ASK_P_PI (0.00180856) |
        ASK_Q_PS (0.00607442) | ASK_Q_PP (0.00367851) | BELIEVE_PS (0.00717886) |
        BELIEVE_PP (0.00434733) | BELIEVE_PI (0.00113035) | BITE_PS (0.0027611) |
        BITE_PP (0.00167205) | BITE_PI (0.00043475) | BUY_PS (0.00960863) |
        BUY_PP (0.00581873) | DRIVE_H_PS (0.0072893) | DRIVE_H_PP (0.00441421) |
        EAT_PS (0.00187755) | EAT_PP (0.00113699) | EXAMINE_PS (0.00209844) |
        EXAMINE_PP (0.00127076) | EXAMINE_PI (0.00033041) | FEEL_PS (0.00607442) |
        FEEL_PP (0.00367851) | FIND_PS (0.0356734) | FIND_PP (0.0216029) |
        FIND_PI (0.00561698) | FOLLOW_PS (0.0146891) | FOLLOW_PP (0.00889531) |
        FOLLOW_PI (0.00231287) | FORGET_M_PS (0.00242977) | FORGET_M_PP (0.0014714) |
        FORGET_M_PI (0.00038258) | FORGET_L_PS (0.00242977) | FORGET_L_PP (0.0014714) |
        FLY_H_PS (0.000993996) | FLY_H_PP (0.000601938) | GET_PS (0.0118175) |
        GET_PP (0.00715637) | GET_PI (0.00186073) | GIVE_PS (0.0461656) |
        GIVE_PP (0.0279567) | GUESS_PS (0.00110444) | GUESS_PP (0.00066882) |
        HAVE_PS (0.0215366) | HAVE_PP (0.013042) | HEAR_PS (0.0130324) |
        HEAR_PP (0.00789208) | HEAR_PI (0.00205202) | HIT_H_PS (0.00474909) |
        HIT_H_PP (0.00287593) | HIT_H_PI (0.000747771) | HIT_O_PS (0.00331332) |
        HIT_O_PP (0.00200646) | HIT_O_PI (0.000521701) | INVOLVE_PS (0.0203217) |
        INVOLVE_PP (0.0123063) | INVOLVE_PI (0.00319976) | KILL_H_PS (0.00618486) |
        KILL_H_PP (0.00374539) | KILL_H_PI (0.000973841) | KILL_A_PS (0.00618486) |
        KILL_A_PP (0.00374539) | KILL_A_PI (0.000973841) | KNOW_P_PS (0.0223097) |
        KNOW_P_PP (0.0135102) | KNOW_P_PI (0.00351278) | KNOW_T_PS (0.0117071) |
        KNOW_T_PP (0.00708949) | LEAVE_I_PS (0.0112653) | LEAVE_I_PP (0.00682196) |
        LEAVE_T_PS (0.0161248) | LEAVE_T_PP (0.00976477) | LEAVE_T_PI (0.00253894) |
        PARK_PS (0.0027611) | PARK_PP (0.00167205) | PLAY_T_PS (0.00662664) |
        PLAY_T_PP (0.00401292) | PUT_PS (0.0188859) | PUT_PP (0.0114368) |
        QUESTION_PS (0.00265066) | QUESTION_PP (0.00160517) | QUESTION_PI (0.00041736) |
        READ_PS (0.00585353) | READ_PP (0.00354475) | SAY_PS (0.00132533) |
        SAY_PP (0.000802584) | SEE_PS (0.0414165) | SEE_PP (0.0250808) |
        SEE_PI (0.00652126) | SHOW_PS (0.0242977) | SHOW_PP (0.014714) |
        SHOW_PI (0.0038258) | TAKE_PS (0.0510251) | TAKE_PP (0.0308995) |
        TAKE_PI (0.00803419) | TELL_S_PS (0.0118175) | TELL_S_PP (0.00715637) |
        TELL_P_PS (0.0055222) | TELL_P_PP (0.0033441) | TELL_P_PI (0.000869501) |
        THINK_PS (0.00949818) | THROW_PS (0.00574309) | THROW_PP (0.00347786) |
        USE_PS (0.0816181) | USE_PP (0.0494258) | WANT_PS (0.00298199) |
        WANT_PP (0.00180581) | WRITE_S_PS (0.0121488) | WRITE_S_PP (0.00735702) |
```

```
        WRITE_P_PS (0.00530131) | WRITE_P_PP (0.00321034) | WRITE_P_PI (0.000834721);

VERB_PR:
        ASK_P_PR (0.0143172) | ASK_Q_PR (0.00550661) | BELIEVE_PR (0.00550661) |
        BITE_PR (0.0132159) | BUY_PR (0.0187225) | DRIVE_H_PR (0.0231278) |
        EAT_PR (0.00440529) | EXAMINE_PR (0.00330396) | FEEL_PR (0.00440529) |
        FIND_PR (0.0473568) | FOLLOW_PR (0.0297357) | FORGET_M_PR (0.00330396) |
        FORGET_L_PR (0.00330396) | FLY_H_PR (0.00440529) | GET_PR (0.00220264) |
        GIVE_PR (0.0539648) | GUESS_PR (0.00220264) | HAVE_PR (0.00330396) |
        HEAR_PR (0.0121145) | HIT_H_PR (0.00991189) | HIT_O_PR (0.00991189) |
        INVOLVE_PR (0.104626) | KILL_H_PR (0.0121145) | KILL_A_PR (0.0121145) |
        KNOW_P_PR (0.0561674) | KNOW_T_PR (0.0418502) | LEAVE_I_PR (0.0242291) |
        LEAVE_T_PR (0.030837) | PARK_PR (0.0110132) | PLAY_T_PR (0.0143172) |
        PUT_PR (0.0231278) | QUESTION_PR (0.00440529) | READ_PR (0.00770925) |
        SAY_PR (0.00660793) | SEE_PR (0.0242291) | SHOW_PR (0.0418502) |
        TAKE_PR (0.0473568) | TELL_S_PR (0.00330396) | TELL_P_PR (0.00330396) |
        THROW_PR (0.00550661) | USE_PR (0.209251) | WANT_PR (0.00330396) |
        WRITE_S_PR (0.0220264) | WRITE_P_PR (0.0165198);

ASK_P_AS:
        TNS_PST asked (0.711755) | TNS_PST was asking (0.0225443) |
        TNS_PST had asked (0.0483092) | TNS_PRS asks (0.10789) |
        TNS_PRS is asking (0.0402576) | TNS_PRS has asked (0.0499195) |
        TNS_FUT will ask (0.0193237);
ASK_P_AP:
        TNS_PST asked (0.711755) | TNS_PST were asking (0.0225443) |
        TNS_PST had asked (0.0483092) | TNS_PRS ask (0.10789) |
        TNS_PRS are asking (0.0402576) | TNS_PRS have asked (0.0499195) |
        TNS_FUT will ask (0.0193237);
ASK_P_AI:
        TNS_PST asked (0.711755) | TNS_PST was asking (0.0225443) |
        TNS_PST had asked (0.0483092) | TNS_PRS ask (0.10789) |
        TNS_PRS am asking (0.0402576) | TNS_PRS have asked (0.0499195) |
        TNS_FUT will ask (0.0193237);
ASK_P_AR:
        asking;
ASK_P_PS:
        TNS_PST was asked (0.637931) | TNS_PST had been asked (0.0344828) |
        TNS_PRS is asked (0.189655) | TNS_PRS has been asked (0.0689655) |
        TNS_FUT will be asked (0.0689655);
ASK_P_PP:
        TNS_PST were asked (0.637931) | TNS_PST had been asked (0.0344828) |
        TNS_PRS are asked (0.189655) | TNS_PRS have been asked (0.0689655) |
        TNS_FUT will be asked (0.0689655);
ASK_P_PI:
        TNS_PST was asked (0.637931) | TNS_PST had been asked (0.0344828) |
        TNS_PRS am asked (0.189655) | TNS_PRS have been asked (0.0689655) |
        TNS_FUT will be asked (0.0689655);
ASK_P_PR:
        TNS_PST asked;

ASK_Q_AS:
        TNS_PST asked (0.711755) | TNS_PST was asking (0.0225443) |
        TNS_PST had asked (0.0483092) | TNS_PRS asks (0.10789) |
        TNS_PRS is asking (0.0402576) | TNS_PRS has asked (0.0499195) |
        TNS_FUT will ask (0.0193237);
ASK_Q_AP:
        TNS_PST asked (0.711755) | TNS_PST were asking (0.0225443) |
        TNS_PST had asked (0.0483092) | TNS_PRS ask (0.10789) |
        TNS_PRS are asking (0.0402576) | TNS_PRS have asked (0.0499195) |
        TNS_FUT will ask (0.0193237);
ASK_Q_AI:
        TNS_PST asked (0.711755) | TNS_PST was asking (0.0225443) |
        TNS_PST had asked (0.0483092) | TNS_PRS ask (0.10789) |
        TNS_PRS am asking (0.0402576) | TNS_PRS have asked (0.0499195) |
        TNS_FUT will ask (0.0193237);
ASK_Q_AR:
        asking;
```

```
ASK_Q_PS:
        TNS_PST was asked (0.637931) | TNS_PST had been asked (0.0344828) |
        TNS_PRS is asked (0.189655) | TNS_PRS has been asked (0.0689655) |
        TNS_FUT will be asked (0.0689655);
ASK_Q_PP:
        TNS_PST were asked (0.637931) | TNS_PST had been asked (0.0344828) |
        TNS_PRS are asked (0.189655) | TNS_PRS have been asked (0.0689655) |
        TNS_FUT will be asked (0.0689655);
ASK_Q_PR:
        TNS_PST asked;
```

D.2 The Penglish lexicon

Each verb and noun in the Penglish language has associated with it a lexical entry which contains a variety of information about the word. Table 5.2 contained a summary of some of the information in the verb entries, including which argument structures could be used by each verb and which thematic roles could be played by modifiers of the verb.

Table D.1, shown below, provides additional information from the verbs' lexical entries. The first four columns give the frequency of occurrence of the verbs in the Penn Treebank (Marcus et al., 1994) with four argument structures: intransitive, transitive, ditransitive using a double object, and ditransitive using a prepositional dative involving *to*.

The values are the total number of occurrences across both the WSJ and Brown corpora. In cases where two values are separated by a slash (200/315), the second value represents the actual measured count from the Treebank and the first value is the one used in the lexicon. There are a few reasons that the frequencies have been edited in this way. One is that multiple senses of verbs have been separated. The frequency analysis of the Treebank is simply based on word form. Distinguishing different verb senses is a very difficult problem. Therefore, when two senses of a verb, like ASK_P versus ASK_Q, were desired, the frequency counts were distributed among the senses as seemed reasonable. For example, ASK_Q cannot be used intransitively, but it can take sentential complements, which ASK_P cannot.

Another reason that frequencies were edited is that strange usages sometimes occur in the Treebank that were not desired in Penglish. For example, the verb *go/went* was used 91 times as a transitive and *have* was used 1,424 times as an intransitive. These are due to expressions like, "*go home*," "*go their respective ways*," and, "*have to do something*," which are not used in Penglish. Finally, some usages which were not found in the Treebank were given small positive frequency counts in cases where they were deemed valid in English despite the fact that they never appeared in the Treebank due to its limited size.

The middle four columns in Table D.1 give the frequency of usage of sentential complements with the verbs. The first two columns are SCs without indirect objects, "asked that we be quiet," and the second two columns are with indirect objects, "asked us that we be quiet." The first and third columns are for SCs marked with *that* or *if* and the others are for reduced SCs. The last four columns are for the occurrence of the verb as the head of passive relative clauses, either marked or reduced, and with and without a by-phrase expressing agency. The reason for these values was to better control the experiments on the main verb/reduced relative ambiguity.

Table D.2 contains the frequency of occurrence of the verbs in the seven active tenses and five passive tenses used in Penglish. Only the relative frequencies of these values matter. The absolute frequencies do not affect the overall frequency of the verbs in Penglish. The overall verb frequency is determined by the argument structure counts given in Table D.1. Because there was no reason to expect one sense of a verb to have a different tense distribution than another sense, at least not in any predictable way, the same values have been used for all senses of the same verb.

The final table of verb information, D.3, gives the frequency with which each verb is modified by a prepositional phrase beginning with a particular preposition, or by an adverb. Although a particular preposition, such as *of*, can express different thematic roles in modifying different verbs, it was assumed that all *of* PPs modifying a particular verb all play the same role. Those roles, as well as the set of nouns that can fill those roles, were also specified in the verbs' lexical entries.

Table D.4 contains the frequency of occurrence of the Penglish nouns as measured in the Penn Treebank. The first two columns give the noun frequencies in singular and plural forms, and the other columns give the frequency with which the nouns are modified by particular types of prepositional phrase. Because the Treebank is mainly composed of news stories and other written language, there is little use of the personal pronouns *I* and *you*. Nevertheless, Penglish

Verb	Intransitive	Transitive	Ditransitive	Prep. Dative	Mrk. SC w/o Obj.	Red. SC w/o Obj.	Mrk. SC w/ Obj.	Red. SC w/ Obj.	Mrk. Pass. RC w/o By	Mrk. Pass. RC w/ By	Red. Pass. RC w/o By	Red. Pass. RC w/ By
ASK_P	331	200/315	0/11	0	0/43	0/2	0/41	0/1	100/153	4/6	12/16	1/2
ASK_Q	0/331	115/315	0/11	0	43	0/2	41	0	53/153	2/6	4/16	1/2
BELIEVE	118	48	0/2	0	184	286	1/0	2	65	0	5	0
BARK	50/4	0	0	0	0	0	0	0	0	0	0	0
BITE	40/13	50/14	0	0/2	0	0	0	0	20/6	5/2	10/0	2/0
BUY	93	904	10	0	0/2	0	0	0	80	7	16	1/0
CONSIST	160	0	0	0	0	0	0	0	0/3	0	0	0
DRIVE_H	60/93	112/122	0/2	0/19	0	0	0	0	44	22	12	9
DRIVE_C	33/93	10/122	0/2	0/19	0	0	0	0	0/44	0/22	0/12	0/9
EAT	64	68	0	0	0	0	0	0	13	4	2	2
EXAMINE	0/13	67	0/2	0/2	0/3	0	0	0	17	2	2	1/0
FEEL	0/284	153	0/3	0	92	106	0	0	50	5	2	2
FIND	0/159	722	0/17	0/3	142	56	4	1/0	311	12	39	4
FISH	30/9	0	0	0	0	0	0	0	0	0	0	0
FOLLOW	166	356	0/3	0/10	0/14	0	0	0	44	89	0	27
FORGET_M	31	21/41	0	0	15	8	0	0/2	21/42	1/0	2/0	1/0
FORGET_L	0/31	20/41	0	0	15	8	0	0/2	21/42	1/0	2/0	1/0
FLY_H	30/111	21	0	0/5	0	0	0	0	7	2	2	2
FLY_P	81/111	0/21	0	0/5	0	0	0	0	0/7	0/2	0/2	0/2
GET	0/923	1000/1164	200/18	0/21	0/4	0	0/4	0	97	10/0	1/0	1/0
GIVE	65	440	683	219	0/2	0	0/2	0	367	51	31	18
GO	1000/2346	0/91	0	0/5	0/4	0/3	0	0	0	0/2	0/2	0
GUESS	25	8	0	0	10	44	0	0	8	2	1/0	1/0
HAVE	0/1424	1000/5185	0/25	0/25	0/3	0	0/8	0/2	194	1/0	2	1/0
HEAR	94	226	0/4	0	15	8	0	0	115	3	9	2
HIT_H	22/32	100/181	0/2	0/3	0	0	0	0	31/51	12/22	2	7
HIT_O	10/32	81/181	0/2	0/3	0	0	0	0	20/51	10/22	2	7
HOPE	186	0/3	0	0	56	89	0	0	0/38	0	0	0
INVOLVE	0/24	229	0	0/2	0/3	0	0	0	184	0	95	0
KILL_H	20	154	0	0	0	0	0	0	49	7	9	2
KILL_A	20	154	0	0	0	0	0	0	49	7	9	2
KNOW_P	0/342	300/456	0/3	0	0/292	0/215	0	0	200/305	2/3	50/87	1/2
KNOW_O	40/342	0/456	0/3	0	0/292	0/215	0	0	0/305	0/3	0/87	0/2
KNOW_T	300/342	156/456	0/3	0	292	215	0	0	105/305	1/3	37/87	1/2
LEAVE_I	214	200/470	0/15	0/14	0/3	0/2	0/4	0	100/242	2/6	20/43	2/7
LEAVE_T	0/214	270/470	0/15	0/14	0/3	0/2	0/4	0	142/242	4/6	23/43	5/7
PARK	5	7	0	0	0	0	0	0	24	1/0	9	1/0
PLAY_T	0/148	299	0/7	0	0	0	0	0	50	10	3	10
PLAY_I	148	0/299	0/7	0	0	0	0	0	0/50	0/10	0/3	0/10
PUT	0/47	670	0/12	0/30	0	0	0/2	0	161	10	16	5
QUESTION	0/13	56	0	0	21	0	0	0	18	6	3	1/0
READ	69	193	20/0	40/9	8	7	1/0	0	48	5	5	2
REALIZE	0/11	0/45	0	0	81	45	0	0	0/27	0/2	0/8	0
SAY	0/2175	51/359	0/2	10/21	151/1055	1037/7260	0	0	10/122	2/0	5	1/0
SEE	306	1040	0/8	0/2	156	21	0/2	0	367	8	20	2
SHOW	0/98	462	21	10	180	47	9	2	204	16	35	3
TAKE	0/255	1000/1926	45	76	0/2	0	0/2	0/2	411	51	28	15
TELL_S	50	300/394	88	7	19	6	185	143	100/147	7/10	1/0	2
TELL_P	0/50	94/394	0/88	0/7	0/19	0/6	0/185	0/143	47/147	3/10	1/0	2
THINK	514	62	0/10	0	109	740	0/2	0/3	84	2	0/7	0
THROW	15	130	3	15	0	0	0	0	49	3	4	1/0
USE	0/98	946	0/2	0/5	0	0	0	0	674	65	156	34
WANT	0/873	324	0/4	0	0	0	0	0	25	2	2	1/0
WISH	0/103	0/12	0/6	0	6	61	0	0	0/5	0	0	0
WRITE_S	0/220	130/230	14	7	30	9	6	2	100/142	10/16	15/27	5/8
WRITE_P	0/220	100/230	0/14	0/7	0/30	0/9	0/6	0/2	42/142	6/16	12/27	3/8

Table D.1: Frequency of verbs with various argument structures and passive relative clauses.

Verb	Simple Past	Past Progressive	Past Perfect	Simple Present	Present Progress.	Present Perfect	Simple Future	Passive Past	Passive Past Perf.	Passive Present	Passive Pres. Perf.	Passive Future
ASK_P	442	14	30	67	25	31	12	37	2	11	4	4
ASK_Q	442	14	30	67	25	31	12	37	2	11	4	4
BELIEVE	92	0	7	382	0	6	1/0	7	0	39	0	0
BARK	20/0	2/0	5/0	10/0	6/2	2/0	2/0	0	0	0	0	0
BITE	28/7	12/3	12/3	20/5	4/0	2	2	5/0	1/0	0	1/0	0
BUY	154	19	22	93	35	25	27	5	1/0	4	5	2
CONSIST	28	0	1/0	73	0	3	7	0	0	0	0	0
DRIVE_H	80	7	9	29	7	8	10	13	3	7	1/0	2
DRIVE_C	80	7	9	29	7	8	10	0/13	0/3	0/7	0/1	0/2
EAT	20	5	7	23	7	5	6	3	3	1/0	1/0	1/0
EXAMINE	20	2	1/0	9	4	7	3	5	1/0	4	1/0	1/0
FEEL	361	7	13	195	9	16	7	11	1/0	9	2	3
FIND	416	2	35	200	19	63	39	56	7	30	30	5
FISH	6/0	2/0	1/0	5/2	5/2	1/0	2/0	0	0	0	0	0
FOLLOW	145	7	6	165	12	15	15	24	0	24	2	5
FORGET_M	18	1/0	16	7	4	14	2	2	2	1/0	2	1/0
FORGET_L	18	1/0	16	7	4	14	2	2	2	1/0	2	1/0
FLY_H	40	5	2	10	4	3	3	2	1/0	1/0	1/0	1/0
FLY_P	40	5	2	10	4	3	3	2	1/0	1/0	1/0	1/0
GET	642	40	26	309	102	164	94	8/0	2/0	0/36	3/0	4/0
GIVE	393	9	53	303	19	65	50	66	16	58	16	14
GO	675	130	5	317	428	2	82	0	0	0	0	0
GUESS	13	1/0	3	48	2	7	1/0	1/0	1/0	0	0	1/0
HAVE	2	19	96	4	41	273	315	1/0	1/0	4	1/0	1/0
HEAR	153	1/0	38	48	6	38	7	12	3	9	4	0
HIT_H	106	5	10	30	2	19	5	21	6	3	8	2
HIT_O	106	5	10	30	2	19	5	21	6	3	8	2
HOPE	48	9	26	188	8	3	1/0	4	0	3	0	0
INVOLVE	39	1/0	2	89	1/0	8	8	15	2	27	7	4
KILL_H	51	1/0	5	15	2	8	5	12	9	4	3	2
KILL_A	51	1/0	5	15	2	8	5	12	9	4	3	2
KNOW_P	425	1/0	35	523	1/0	34	18	22	3	71	4	6
KNOW_O	425	1/0	35	523	1/0	34	18	22	3	71	4	6
KNOW_T	425	1/0	35	523	1/0	34	18	22	3	71	4	6
LEAVE_I	245	10	50	107	24	56	15	19	3	38	6	2
LEAVE_T	245	10	50	107	24	56	15	19	3	38	6	2
PARK	7	1/0	2	1/0	1/0	2	1/0	5	2	4	0	0
PLAY_T	106	15	7	88	12	26	13	7	1/0	5	1/0	2
PLAY_I	106	15	7	88	12	26	13	7	1/0	5	1/0	2
PUT	215	5	23	111	23	30	14	28	6	18	12	2
QUESTION	25	2	3	23	7	3	2	6	2	1/0	1/0	2
READ	49	5	11	50	7	18	2	4	1/0	3	3	1/0
REALIZE	69	1/0	3	39	3	7	4	0/3	0	0/4	0/2	0/3
SAY	8968	21	78	3405	32	139	12	28	3	43	7	0
SEE	429	8	96	257	18	139	45	23	5	22	4	7
SHOW	263	13	14	263	16	71	16	14	3	33	7	6
TAKE	698	28	67	290	79	120	144	63	12	39	13	11
TELL_S	483	5	38	101	26	26	20	47	6	24	6	3
TELL_P	483	5	38	101	26	26	20	47	6	24	6	3
THINK	439	27	23	502	19	20	9	13	2	26	1/0	2
THROW	62	3	9	22	1/0	6	3	16	2	5	1/0	1/0
USE	224	10	14	183	34	42	28	69	3	158	24	46
WANT	319	1/0	13	606	2	9	11	2	1/0	3	1/0	1/0
WISH	54	1/0	3	89	2	5	1/0	2	1/0	1/0	1/0	1/0
WRITE_S	253	11	25	76	4	22	6	18	1/0	28	9	4
WRITE_P	253	11	25	76	4	22	6	18	1/0	28	9	4

Table D.2: Frequency of verbs with various tenses, aspects, and voices.

Verb	*of PP*	*for PP*	*with PP*	*to PP*	*in PP*	*on PP*	*Adverb*
ASK_P	6	128	2	0	11	2	20/73
ASK_Q	6	128	2	0	11	2	20/73
BELIEVE	0	0	0	0	46	4	5/30
BARK	0	0	0/2	0	0	0	5/0
BITE	0	0	2	2	0	3	3
BUY	0	110	9	2	69	22	50/137
CONSIST	160/148	0	0/2	0	0/12	0	19
DRIVE_H	2	6	8	36	31	6	20/82
DRIVE_C	2	6	8	25/36	31	6	10/82
EAT	2	3	5	0	10	2	17
EXAMINE	0	9	3	0	6	0	10
FEEL	0	19	6	3	24	8	4/60
FIND	3	15	12	5	215	20	10/72
FISH	0	2	2	0	2	0	2
FOLLOW	0	12	16	13	28	9	51
FORGET_M	0	4	0	0	3	0	13
FORGET_L	0	4	0	0	3	0	13
FLY_H	2	8	5	24	11	11	31
FLY_P	2	8	5	24	11	11	31
GET	7	56	41	93	117	75	414
GIVE	0	36	7	304	74	10	88
GO	4	86	113	421	88	69	714
GUESS	0	0	0	0	0	3	9
HAVE	11	127	101	32	383	131	836
HEAR	33	9	3	3	21	10	40
HIT_H	2	4	19	0	29	11	64
HIT_O	2	4	19	0	29	11	64
HOPE	2	18	3	3	3	0	17
INVOLVE	0	2	6	2	101	2	31
KILL_H	0	3	2	0	18	5	16
KILL_A	0	3	2	0	18	5	16
KNOW_P	0/40	30	5	10	28	2	164
KNOW_O	40	0/30	0/5	0/10	0/28	0/2	4/164
KNOW_T	0/40	30	5	10	28	2	164
LEAVE_I	5	55	72	38	57	17	88
LEAVE_T	5	55	72	38	57	17	88
PARK	0	0	0	0	13	3	6
PLAY_T	1/0	32	55	4	85	21	74
PLAY_I	1/0	32	55	4	85	21	74
PUT	0/2	0/29	0/19	0/56	388/169	474/184	177
QUESTION	0	0	2	0	8	0	12
READ	5	3	10	10	20	4	38
REALIZE	2	4	4	2	7	5	19
SAY	33	16	32	95	175	14	316
SEE	3	21	17	17	146	34	179
SHOW	0	14	11	12	112	20	72
TAKE	74	61	62	133	192	62	314
TELL_S	42	4	5	9	19	9	59
TELL_P	42	4	5	9	19	9	59
THINK	221	10	6	4	18	3	119
THROW	2	7	3	12	14	14	40
USE	2	139	13	25	220	31	153
WANT	0	17	3	0	12	3	43
WISH	2	4	2	3	0	0	11
WRITE_S	13	33	15	51	70	27	53
WRITE_P	13	33	15	51	70	27	53

Table D.3: Frequency of verbs with various PPs and adverbs.

Noun	Singular	Plural	of PP	for PP	by PP	with PP	on PP
I	1000/?	200/?	0	0	0	0	0
YOU	0	600/?	0	0	0	0	0
BOY	249	142	8	0	0	4/7	3
GIRL	225	147	5	0	0	4/6	2
LAWYER	120	180	0	23	0	4/6	1/0
TEACHER	93	95	16	1	0	1	1
REPORTER	39	82	2/0	1	0	1/0	1/0
PLAYER	57	104	2	1	0	1	2
COP	89/15	60/16	1	0	0	1/0	2
FATHER	229	15	18	0	0	3/0	1/0
MOTHER	256	35	18	0	0	3/2	1/0
MANAGER	276	230	84	24	0	2/1	0
OWNER	95	119	48	0	0	1/0	0
CAT	22	19	0	0	0	1/0	1/0
DOG	64	77	3	0	0	2/0	2
BIRD	17	51	1	0	0	1/0	1/0
FISH	23	20/0	1	0	0	1/0	1/0
BAT_A	19	8	0	0	0	1/0	1/0
QUESTION	367	237	91	20/3	0	1	10/1
ANSWER	144	63	2	4	2/0	0	3
TEST	136	121	29	8	1	4	3
STORY	220	98	50	1	1	3	7
BOOK	222	142	22	2/1	3	3	3/13
APPLE	11	22	1	1/0	0	0	1/0
FOOD	140	0	1	9	1	1	1
CAR	337	266	4	5/2	0	7	2
PLANE	119	60	23	1	0	2	0
BALL	90	23	4	1	0	1/0	1/0
BAT_O	19	8	0	1/0	0	1/0	1/0
BASEBALL_B	58	2	0	1	0	1/0	1/0
BASEBALL_G	58	0	0	1	0	0	0
TRUMPET	40/4	10/0	0	1/0	0	0	1/0
EVIDENCE	318	0	71	3	0	0	6
SOMETHING	300/609	0	20/41	2	1/0	0	1/0
BINOCULARS	0	20/3	0	1/0	0	1/0	1/0
KNIFE	63	7	1	1/0	0	0	1/2
PEN	32/16	6/3	3	1/0	0	1/0	1/0
PICTURE	187	81	76	3	1/0	3	1
TABLE	218	60	9	1	1	1	3
FLOOR	210	23	28	1	0	0	3
HOUSE	504	151	67	6	0	6	9
PARK	57	22	2	1/0	0	3	1
ZOO	12/6	1/0	1	1/0	0	1/0	0
SCHOOL_I	308	239	53	11	0	4	0
SCHOOL_G	308	0	42	8	0	1	0

Table D.4: Frequency of nouns in singular and plural form and with various PPs.

is intended to be mainly a model of spoken language, in which those pronouns are quite common. Therefore, the frequencies of the personal pronouns were estimated using the CHILDES database (MacWhinney, 1999).

References

Abney, S. P. (1989). A computational model of human parsing. *Journal of Psycholinguistic Research*, *18*, 129–144. [pg. 60]

Adams, B. C., Clifton, C., Jr., & Mitchell, D. C. (1991). *Lexical guidance in sentence processing.* Poster presented at the meeting of the Psychonomics Society, San Francisco. [pg. 57]

Allen, R. B. (1987). Several studies on natural language and back-propagation. In *Proceedings of the International Conference on Neural Networks* (pp. II/335–341). San Diego. [pg. 29]

Allen, R. B. (1988). Sequential connectionist networks for answering simple questions about a microworld. In *Proceedings of the 10th annual conference of the Cognitive Science Society* (pp. 489–495). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 23]

Altmann, G., & Steedman, M. (1988). Interaction with context during human sentence processing. *Cognition*, *30*, 191–238. [pp. 63–64,66]

Angluin, D. (1988). *Identifying languages from stochastic examples* (Tech. Rep. No. YALEU/DCS/RR-614). New Haven, CT: Yale University, Department of Computer Science. [pg. 9]

Baird, R., & Koslick, J. D. (1974). Recall of grammatical relations within clause-containing sentences. *Journal of Psycholinguistic Research*, *3*(2), 165–171. [pp. 36,38,236]

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*, 157–166. [pg. 128]

Berg, G. (1992). A connectionist parser with recursive sentence structure and lexical disambiguation. In *Proceedings of the 10th National Conference on Artificial Intelligence* (pp. 32–37). San Jose, CA: AAAI. [pg. 21]

Berwick, R. C. (1985). *The acquisition of syntactic knowledge.* Cambridge, MA: MIT Press. [pg. 8]

Bever, T. G. (1970). The cognitive basis for linguistic structure. In J. R. Hayes (Ed.), *Cognitive Development of Language.* New York: John Wiley. [pp. 7,44]

Bickerton, D. (1984). The language bioprogram hypothesis. *Behavioral and Brain Sciences*, *7*, 173–221. [pg. 6]

Blaubergs, M. S., & Braine, M. D. S. (1974). Short-term memory limitations on decoding self-embedded sentences. *Journal of Experimental Psychology*, *102*(4), 745–748. [pp. 32,35,38–39,41,235]

Blumenthal, A. L. (1966). Observations with self-embedded sentences. *Psychonomic Science*, *6*(10), 453–454. [pp. 38,40]

Bock, J. K. (1986). Syntactic persistence in language production. *Cognitive Psychology*, *18*, 355–387. [pp. 67–69,273–274]

Bock, K. (1990). Structure in language: Creating form in talk. *American Psychologist*, *45*, 1221–1236. [pg. 66]

Bock, K. (1995). Sentence production: From mind to mouth. In J. Miller & P. Eimas (Eds.), *Speech, language and communication* (pp. 181–216). Academic Press. [pp. 249,268]

Bock, K., & Eberhard, K. M. (1993). Meaning, sound and syntax in English number agreement. *Language and Cognitive Processes*, *8*, 57–99. [pp. 268–270,284]

Bock, K., & Griffin, Z. M. (2000). The persistence of structural priming: Transient activation or implicit learning? *Journal of Experimental Psychology: General*, *129*, 177–192. [pp. 68,274–275,284]

Bock, K., & Loebell, H. (1990). Framing sentences. *Cognition*, *35*, 1–39. [pp. 66–69,273,275,277]

Bock, K., Loebell, H., & Morey, R. (1989). *From conceptual roles to structural relations: Bridging the syntactic cleft.* Paper presented at the meeting of the Psychonomic Society., Atlanta, GA. [pp. 67,69,73,273]

Bock, K., & Miller, C. A. (1991). Broken agreement. *Cognitive Psychology*, *23*, 45–93. [pp. 268–271,284]

Branigan, H. P., Pickering, M. J., & Cleland, A. A. (2000). Syntactic coordination in dialogue. *Cognition*, *75*, B13–B25. [pp. 68,274]

Britt, M. A. (1994). The interaction of referential ambiguity and argument structure in the parsing of prepositional phrases. *Journal of Memory and Language*, *33*, 251–283. [pp. 60–61,64–66,96–97]

Britt, M. A., Perfetti, C. A., Garrod, S., & Rayner, K. (1992). Parsing in discourse: Context effects and their limits. *Journal of Memory and Language*, *31*, 293–314. [pp. 60,64−66]

Burgess, C. (1991). *Interaction of semantic, syntactic and visual factors in syntactic ambiguity resolution.* Unpublished doctoral dissertation, University of Rochester, Rochester, NY. [pg. 46]

Burgess, C., & Tanenhaus, M. K. (1992). *Semantic, syntactic and visual factors in syntactic ambiguity resolution.* Unpublished manuscript. [pg. 47]

Burgess, C., Tanenhaus, M. K., & Hoffman, M. (1994). Parafoveal and semantic effects on syntactic ambiguity resolution. In *Proceedings of the 16th annual conference of the Cognitive Science Society* (pp. 96–99). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 46−47,192]

Caplan, D., & Hildebrandt, N. (1988). *Disorders of syntactic comprehension.* Cambridge, MA: MIT Press. [pp. 171,280]

Chalmers, D. J. (1990). Syntactic transformations on distributed representations. *Connection Science*, *2*, 53–62. [pg. 29]

Chang, F., Dell, G. S., Bock, K., & Griffin, Z. M. (2000). Structural priming as implicit learning: A comparison of models of sentence production. *Journal of Psycholinguistic Research*, *29*, 217–229. [pp. 68,251,274]

Charniak, E., & Santos, E. (1987). A connectionist context-free parser which is not context-free, but then it is not really connectionist either. In *Proceedings of the 9th annual conference of the Cognitive Science Society* (pp. 70–77). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 19]

Chater, N., & Conkey, P. (1992). Finding linguistic structure with recurrent neural networks. In *Proceedings of the 14th annual conference of the Cognitive Science Society* (pp. 402–407). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 26]

Chomsky, N. (1959). A review of B.F. Skinner's *Verbal Behavior*. *Language*, *35*, 26–58. [pg. 3]

Chomsky, N. (1965). *Aspects of the theory of syntax.* Cambridge, MA: MIT Press. [pg. 296]

Chomsky, N. (1975). *Reflections on language.* New York: Pantheon Books. [pg. 296]

Christiansen, M. H. (1994). *Infinite languages, finite minds: Connectionism, learning, and linguistic structure.* Unpublished doctoral dissertation, University of Edinburgh. [pg. 26]

Christiansen, M. H., & Chater, N. (1999a). Connectionist natural language processing: the state of the art. In M. H. Christiansen, N. Chater, & M. S. Seidenberg (Eds.), *Special Issue of Cognitive Science: Connectionist Models of Human Language Processing: Progress and Prospects* (Vol. 23). Cognitive Science. [pp. i,17,144]

Christiansen, M. H., & Chater, N. (1999b). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, *23*, 157–205. [pp. 12,26,251,284,314]

Christianson, K., Hollingworth, A., Halliwell, J. F., & Ferreira, F. (2001). Thematic roles assigned along the garden path linger. *Cognitive Psychology*, *42*, 368–407. [pp. 229−231,233,283]

Clifton, C., Jr. (1993). Thematic roles in sentence parsing. *Canadian Journal of Experimental Psychology*, *47*, 222–246. [pg. 234]

Clifton, C., Jr., Speer, S., & Abney, S. P. (1991). Parsing arguments: Phrase structure and argument structure as determinants of initial parsing decisions. *Journal of Memory and Language*, *31*, 251–271. [pg. 60−61]

Cottrell, G. W. (1985a). Connectionist parsing. In *Proceedings of the 7th annual conference of the Cognitive Science Society* (pp. 201–211). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 18]

Cottrell, G. W. (1985b). *A connectionist approach to word sense disambiguation.* Unpublished doctoral dissertation, Department of Computer Science, University of Rochester, Rochester, NY. [pg. 18]

Crain, S., & Steedman, M. (1985). On not being led up the garden path: The use of context by the psychological syntax processor. In D. R. Dowty, L. Karttunen, & A. M. Zwicky (Eds.), *Natural language processing* (pp. 320–358). Cambridge, UK: Cambridge University Press. [pp. 62,65−66]

Crystal, D. (1997). *The cambridge encyclopedia of language.* Cambridge, UK: Cambridge University Press. [pg. 7]

Cuetos, F., & Mitchell, D. C. (1988). Cross-linguistic differences in parsing: Restrictions on the use of the Late Closure strategy in Spanish. *Cognition*, *30*, 73–105. [pp. 59,73]

Daneman, M., & Carpenter, P. A. (1980). Individual differences in comprehending and producing words in context. *Journal of Verbal Learning and Verbal Behavior*, *19*, 450–466. [pg. 49]

Dell, G. S. (1986). A spreading activation theory of retrieval in language production. *Psychological Review*, *93*, 283–321. [pg. 27]

Dell, G. S., Chang, F., & Griffin, Z. M. (1999). Connectionist models of language production: Lexical access and grammatical encoding. In M. H. Christiansen, N. Chater, & M. S. Seidenberg (Eds.), *Special Issue of Cognitive Science: Connectionist Models of Human Language Processing: Progress and Prospects* (Vol. 23). Cognitive Science. [pp. 28,68,274,276]

Dell, G. S., Juliano, C., & Govindjee, A. (1993). Structure and context in language production: A theory of frame constraints in phonological speech errors. *Cognitive Science*, *17*, 149–195. [pg. 27]

Diederich, J. (1989). *Spreading activation and connectionist models for natural language processing* (Tech. Rep. No. TR-89-008). Berkeley, CA: International Computer Science Institute. [pg. 17]

Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, *67*, 547–619. [pg. 121]

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211. [pp. 10,12,21−22,24,127−128]

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, *7*, 195–225. [pp. i,12,24−26,116,148,152,280,284]

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, *48*, 71–99. [pg. 24−25]

Fahlman, S. E., Hinton, G. E., & Sejnowski, T. J. (1983). Massively parallel architectures for AI: NETL, Thistle, and Boltzmann machines. In *Proceedings of the National Conference on Artificial Intelligence* (pp. 109–113). Washington. [pg. 19]

Fanty, M. (1985). *Context-free parsing in connectionist networks* (Tech. Rep. No. TR-174). Rochester, NY: University of Rochester, Computer Science Department. [pg. 18−19]

Fanty, M. (1994). Context-free parsing in connectionist networks. In G. Adriaens & U. Hahn (Eds.), *Parallel natural language processing* (pp. 211–237). Norwood, NJ: Ablex Publishing. [pg. 18]

Ferreira, F., Christianson, K., & Hollingworth, A. (2001). Misinterpretations of garden-path sentences: Implications for models of sentence processing and reanalysis. *Journal of Psycholinguistic Research*, *30*, 3–20. [pg. 229]

Ferreira, F., & Henderson, J. M. (1990). Use of verb information in syntactic parsing: Evidence from eye movements and word-by-word self-paced reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *16*, 555–568. [pp. 53−54,56,207]

Ferreira, F., & Henderson, J. M. (1991). Recovery from misanalysis of garden-path sentences. *Journal of Memory and Language*, *30*, 724–745. [pp. 57−59,225−227,233,283]

Ferreira, F., & Henderson, J. M. (1993). Reading processes during syntactic analysis and reanalysis. *Canadian Journal of Experimental Psychology*, *47*, 247–275. [pg. 58]

Ferriera, F., & Clifton, C., Jr. (1986). The independence of syntactic processing. *Journal of Memory and Language*, *25*, 348–368. [pp. 44,46−47,49,60,62,64−66,191−192]

Fodor, J. A. (1983). *Modularity of mind.* Cambridge, MA: MIT Press. [pg. 18]

Fodor, J. A., & Garrett, M. (1967). Some syntactic determinants of sentential complexity. *Perception and Psychophysics*, *2*(7), 289–296. [pp. 33,41]

Ford, M. (1983). A method for obtaining measures of local parsing complexity throughout sentences. *Journal of Verbal Learning and Verbal Behavior*, *22*, 203–218. [pp. 37−38,236]

Foss, D. J., & Lynch, R. H. (1969). Decision processes during sentence comprehension: Effects of surface structure on decision times. *Perception and Psychophysics*, *5*, 145–148. [pg. 42]

Francis, W. N., & Kucera, H. (1979). *Manual of information to accompany a standard corpus of present-day edited American English.* Providence, RI: Brown University, Department of Linguistics. [pg. 28]

Frazier, L. (1979). *On comprehending sentences: Syntactic parsing strategies.* Bloomington, IN: Indiana University Linguistics Club. [pp. 18,52]

Frazier, L. (1987). Theories of sentence processing. In J. L. Garfield (Ed.), *Modularity in knowledge representation and natural language understanding.* Cambridge, MA: MIT Press. [pg. 52]

Frazier, L., & Fodor, J. D. (1978). The sausage machine: A new two stage parsing model. *Cognition*, *6*, 291–324. [pg. 189]

Frazier, L., & Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, *14*, 178–210. [pp. 57,219,224]

Garnsey, S. M., Pearlmutter, N. J., Myers, E., & Lotocky, M. A. (1997). The contributions of verb bias and plausibility to the comprehension of temporarily ambiguous sentences. *Journal of Memory and Language*, *37*, 58–93. [pp. 55−56,208−209,211,213−214]

Gasser, M., & Dyer, M. G. (1988). Sequencing in a connectionist model of language processing. In *COLING Budapest: Proceedings of the 12th International Conference on Computational Linguistics* (pp. 185–190). Budapest. [pg. 27]

Gasser, M. E. (1988). *A connectionist model of sentence generation in a first and second language.* Unpublished doctoral dissertation, Computer Science Department, University of California, Los Angeles, CA. (TP UCLA-AI-88-13) [pg. 27]

Gibson, E. (1991). *A computational theory of human linguistic processing: Memory limitations and processing breakdowns.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA. [pp. 7,33,296]

Gibson, E. (1997). *Linguistic complexity: Locality of syntactic dependencies* (Tech. Rep.). Cambridge, MA: MIT. [pg. 33]

Gibson, E. (1998). Linguistic complexity: Locality of syntactic dependencies. *Cognition*, *68*, 1–76. [pp. 32,43]

Gibson, E. (2000). The dependency locality theory: A distance-based theory of linguistic complexity. In A. Marantz, Y. Miyashita, & W. O'Neil (Eds.), *Image, language, brain: Papers from the first mind articulation project symposium.* Cambridge, MA: MIT Press. [pp. 33,146,297]

Gibson, E., Desmet, T., Watson, D., Grodner, D., & Ko, K. (in press). *Reading relative clauses in English.* Manuscript submitted for publication. [pp. 37–38,235–237,242,281]

Gibson, E., Pearlmutter, N., Canseco-Gonzalez, E., & Hickok, G. (1996). Recency preference in the human sentence processing mechanism. *Cognition*, *59*, 23–59. [pp. 59,73]

Gibson, E., & Pearlmutter, N. J. (1994). A corpus-based analysis of psycholinguistic constraints on prepositional-phrase attachment. In J. C. Clifton, L. Frazier, & K. Rayner (Eds.), *Perspectives on sentence processing* (pp. 181–198). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 59,73]

Gibson, E., & Pearlmutter, N. J. (2000). Distinguishing serial and parallel parsing. *Journal of Psycholinguistic Research*, *29*, 231–240. [pg. 5]

Gibson, E., & Thomas, J. (1995). The processing complexity of English center-embedded and self-embedded structures. In *Proceedings of the NELS 26 Processing Workshop, Fall 1995.* Cambridge, MA. [pp. 32,40–41,70,94]

Gibson, E., & Thomas, J. (1997). *The complexity of nested structures in English: Evidence for the syntactic prediction locality theory of linguistic complexity* (Tech. Rep.). Cambridge, MA: MIT. [pp. 32,40–42,51,246]

Gibson, E., & Warren, T. (1998). *Discourse reference and syntactic complexity* (Tech. Rep.). Cambridge, MA: MIT. [pp. 43,92,246]

Giles, C. L., Horne, B. G., & Lin, T. (1995). Learning a class of finite state machines with a recurrent neural network. *Neural Networks*, *8*, 1359–1365. [pg. 12]

Glanzer, M., & Cunitz, A. R. (1966). Two storage mechanisms in free recall. *Journal of Verbal Learning and Verbal Behavior*, *5*, 351–360. [pp. 137,163]

Goldowsky, B. N., & Newport, E. L. (1993). Modeling the effects of processing limitations on the acquisition of morphology: the less is more hypothesis. In E. Clark (Ed.), *The proceedings of the 24th annual Child Language Research Forum* (pp. 124–138). Stanford, CA: Center for the Study of Language and Information. [pg. 25]

Grodner, D., Gibson, E., & Watson, D. (in press). *The effect of discourse contrast on syntactic processing: Evidence for strong interaction in sentence comprehension.* Manuscript submitted for publication. [pg. 235]

Hahn, U., & Adriaens, G. (1994). Parallel natural language processing: Background and overview. In G. Adriaens & U. Hahn (Eds.), *Parallel natural language processing* (pp. 1–134). Norwood, NJ: Ablex Publishing. [pg. 17]

Hakes, D. S., & Cairns, H. S. (1970). Sentence comprehension and relative pronouns. *Perception and Psychophysics*, *8*(1), 5–8. [pg. 42]

Hakes, D. S., Evans, J. S., & Brannon, L. L. (1976). Understanding sentences with relative clauses. *Memory and Cognition*, *4*(3), 283–290. [pp. 36–38,42,236]

Hanson, S. J., & Kegl, J. (1987). Parsnip: A connectionist network that learns natural language grammar from exposure to natural language sentences. In *Proceedings of the 9th annual conference of the Cognitive Science Society* (pp. 106–119). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 28–29]

Hare, M., & Elman, J. (1995). Learning and morphological change. *Cognition*, *56*, 61–98. [pg. 6]

Harley, T. A. (1993). Phonological activation of semantic competitors during lexical access in speech production. *Language and Cognitive Processes*, *8*, 291–309. [pg. 27]

Harm, M. W., Thornton, R., & MacDonald, M. C. (2000). A distributed, large scale connectionist model of the interaction of lexical and semantic constraints in syntactic ambiguity resolution [Abstract]. In *Proceedings of the 13th annual CUNY Conference on Human Sentence Processing.* La Jolla, CA. [pg. 21]

Henderson, J. B. (1990). *Structure unification grammar: A unifying framework for investigating natural language* (Tech. Rep. No. MS-CIS-90-94). Philadelphia, PA: University of Pennsylvania. [pg. 21]

Henderson, J. B. (1994a). Connectionist syntactic parsing using temporal variable binding. *Journal of Psycholinguistic Research*, *23*(5), 353–379. [pp. 21,27]

Henderson, J. B. (1994b). *Description based parsing in a connectionist network.* Unpublished doctoral dissertation, University of Pennsylvania, Philadelphia, PA. [pg. 21]

Henderson, J. B. (1996). A connectionist architecture with inherent systematicity. In *Proceedings of the 18th annual conference of the Cognitive Science Society* (pp. 574–579). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 21]

Henderson, J. B., & Lane, P. C. R. (1998). A connectionist architecture for learning to parse. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th annual meeting of the Association for Computational Linguistics (COLING-ACL'98).* University of Montreal, Canada. [pg. 21]

Hinton, G. E. (1981). Implementing semantic networks in parallel hardware. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel models of associative memory* (pp. 161–187). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 22]

Hinton, G. E., & Anderson, J. A. (1981). *Parallel models of associative memory.* Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 10]

Hinton, G. E., & Shallice, T. (1991). Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological Review*, *98*, 74–95. [pg. 128]

Holmes, V. M. (1973). Order of main and subordinate clauses in sentence perception. *Journal of Verbal Learning and Verbal Behavior*, *12*, 285–293. [pp. 36,38,235]

Holmes, V. M. (1987). Syntactic parsing: In search of the garden path. In M. Coltheart (Ed.), *Attention and performance xii: The psychology of reading* (pp. 587–599). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 49,52–53,56,205,207,209,213–214]

Holmes, V. M., Kennedy, A., & Murray, W. S. (1987). Syntactic structure and the garden path. *Quarterly Journal of Experimental Psychology*, *39A*, 277–293. [pp. 52–53,56,205,209,213]

Holmes, V. M., & O'Regan, J. K. (1981). Eye fixation patterns during the reading of relative-clause sentences. *Journal of Verbal Learning and Verbal Behavior*, *20*, 417–430. [pg. 37–38]

Holmes, V. M., Stowe, L., & Cupples, L. (1989). Lexical expectations in parsing complement-verb sentences. *Journal of Memory and Language*, *28*, 668–689. [pp. 52–53,56,207,209,214,216–217,282]

Hopcroft, J. E., & Ullman, J. D. (1979). *Introduction to automata theory, languages, and computation.* Reading, MA: Addison-Wesley Publishing. [pp. 309,321–322,324]

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, *79*, 2554–2558. [pg. 128]

Horning, J. J. (1969). *A study of grammatical inference.* Unpublished doctoral dissertation, Stanford University. [pg. 9]

Howells, T. (1988). VITAL: A connectionist parser. In *Proceedings of the 10th annual conference of the Cognitive Science Society* (pp. 18–25). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 19]

Hudgins, J. C., & Cullinan, W. L. (1978). Effects of sentence structure on sentence elicited imitation responses. *Journal of Speech and Hearing Research*, *21*, 809–819. [pp. 36,236]

Huffman, D. A. (1952). A method for the construction of minimum redundancy codes. *Proceedings of the Institute of Radio Engineers*, *40*, 1098–1101. [pg. 135]

Jain, A. N., & Waibel, A. H. (1990). Incremental parsing by modular recurrent connectionist networks. In D. Touretzky (Ed.), *Advances in neural information processing systems 2* (pp. 364–371). San Mateo, CA: Morgan Kaufmann. [pg. 20–21]

Johnson-Laird, P. N. (1983). *Mental models.* Cambridge, MA: Harvard University Press. [pg. 5]

Juliano, C., & Tanenhaus, M. K. (1993). Contingent frequency effects in syntactic ambiguity resolution. In *Proceedings of the 15th annual conference of the Cognitive Science Society* (p. 593-598). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 54–56,95–96,106]

Juliano, C., & Tanenhaus, M. K. (1994). A constraint-based lexicalist account of the subject/object attachment preference. *Journal of Psycholinguistic Research*, *23*, 459–471. [pg. 55]

Jurafsky, D. (1996). A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science*, *20*, 137–194. [pg. 291]

Just, M. A., & Carpenter, P. A. (1980). The psychology of reading and language acquisition. *Psychological Review*, *87*, 329–354. [pg. 147]

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, *99*(1), 122–149. [pp. 49–50,146]

Kalita, J., & Shastri, L. (1987). Generation of simple sentences in English using the connectionist model of computation. In *Proceedings of the 9th annual conference of the Cognitive Science Society* (pp. 555–565). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 27]

Kalita, J., & Shastri, L. (1994). A connectionist approach to generation of simple sentences and word choice. In G. Adriaens & U. Hahn (Eds.), *Parallel natural language processing* (pp. 395–420). Norwood, NJ: Ablex Publishing. [pg. 27]

Kimball, J. (1973). Seven principles of surface structure parsing in natural languages. *Cognition*, *2*, 15–47. [pp. 7,33]

King, J., & Just, M. A. (1991). Individual differences in syntactic processing: the role of working memory. *Journal of Memory and Language*, *30*, 580–602. [pp. 37–38,146,236]

Kohonen, T. (1984). *Self-organization and associative memory.* New York: Springer-Verlag. [pg. 10]

Konieczny, L. (1996). *Human sentence processing: A semantics-oriented parsing approach.* Unpublished doctoral dissertation, Albert-Ludwigs University, Freiberg. [pg. 60]

Kucera, H., & Francis, W. N. (1967). *Computational analysis of present-day American English.* Providence, RI: Brown University Press. [pg. 75]

Kukich, K. (1987). Where do phrases come from: Some preliminary experiments in connectionist phrase generation. In G. Kempen (Ed.), *Natural language generation: New results in artificial intelligence, psychology and linguistics* (pp. 405–421). Boston, Dordrecht: Martinus Nijhoff Publishers. [pg. 28]

Kwasny, S. C., & Faisal, K. A. (1990). Connectionism and determinism in a syntactic parser. *Connection Science*, *2*, 63–82. [pg. 20]

Lane, P. C. R., & Henderson, J. B. (1998). Simple synchrony networks: Learning to parse natural language with temporal synchrony variable binding. In *Icann.* Skövde, Sweden. [pg. 21]

Larkin, W., & Burns, D. (1977). Sentence comprehension and memory for embedded structure. *Memory and Cognition*, *5*(1), 17–22. [pp. 39–40,42–43]

Levelt, W. J. M. (1993). *Speaking.* Cambridge, MA: MIT Press. [pg. 153]

Lewis, R. L. (1993). *An architecturally-based theory of human sentence comprehension.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA. [pp. 7,33]

Lewis, R. L. (1996). Interference in short-term memory: The magical number two (or three) in sentence processing. *Journal of Psycholinguistic Research*, *25*(1), 93–115. [pg. 7]

Lewis, R. L. (1998). Reanalysis and limited repair parsing: Leaping off the garden path. In J. D. Fodor & F. Ferreira (Eds.), *Reanalysis in sentence processing* (pp. 247–284). Boston: Kluwer Academic. [pg. 146]

Lewis, R. L. (2000a). Falsifying serial and parallel parsing models: Empirical conundrums and an overlooked paradigm. *Journal of Psycholinguistic Research*, *29*, 241–248. [pg. 74]

Lewis, R. L. (2000b). Specifying architectures for language processing: Process, control, and memory in parsing and interpretation. In M. Crocker, M. Pickering, & C. Clifton, Jr. (Eds.), *Mechanisms for language processing.* Cambridge, UK: Cambridge University Press. [pg. 1]

Luce, D. R. (1986). *Response times.* New York: Oxford University Press. [pg. 138]

MacDonald, M. C. (1994). Probabilistic constraints and syntactic ambiguity resolution. *Language and Cognitive Processes*, *9*, 157–201. [pp. 46,48,50,56,201–202,204,282]

MacDonald, M. C. (1996). Representation and activation in syntactic processing. In T. Inui & J. L. McClelland (Eds.), *Attention and performance xvi: Information integration in perception & communication* (pp. 433–453). Cambridge, MA: MIT Press. [pg. 50]

MacDonald, M. C., & Christiansen, M. H. (in press). Reassessing working memory: A comment on Just & Carpenter (1992) and Waters & Caplan (1996). *Psychological Review*. [pp. 144,297]

MacDonald, M. C., Just, M. A., & Carpenter, P. A. (1992). Working memory constraints on the processing of syntactic ambiguity. *Cognitive Psychology*, *24*, 56–98. [pp. 49,196]

MacDonald, M. C., Pearlmutter, N. J., & Seidenberg, M. S. (1994a). Lexical nature of syntactic ambiguity resolution. *Psychological Review*, *101*(4), 676–703. [pp. 6,45,49,62,152,189,202,284]

MacDonald, M. C., Pearlmutter, N. J., & Seidenberg, M. S. (1994b). Syntactic ambiguity resolution as lexical ambiguity resolution. In J. C. Clifton, L. Frazier, & K. Rayner (Eds.), *Perspectives on sentence processing* (pp. 123–153). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 44,48–50,61]

MacWhinney, B. (1999). *The CHILDES database.* Mahwah, NJ: Lawrence Erlbaum Associates. [pp. 106,358]

MacWhinney, B., & Pléh, C. (1988). The processing of restrictive relative clauses in Hungarian. *Cognition*, *29*, 95–141. [pp. 136,240]

Marcus, G. F. (1993). Negative evidence in language acquisition. *Cognition*, *46*, 53–85. [pg. 8]

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., & Schasberger, B. (1994). The Penn Treebank: annotating predicate argument structure. In *Proceedings of the human language technology workshop.* Morgan Kaufmann Publishers Inc. [pp. 69,75,327,354]

Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, *19*. [pp. 5,75,327]

Marcus, M. P. (1980). *A theory of syntactic recognition for natural language.* Cambridge, MA: MIT Press. [pg. 20]

Marks, L. E. (1968). Scaling of grammaticalness of self-embedded English sentences. *Journal of Verbal Learning and Verbal Behavior*, *7*, 965–967. [pp. 36,38–39,41,235]

Marslen-Wilson, W. (1973). Linguistic structure and speech shadowing at very short latencies. *Nature*, *244*, 522–533. [pg. 153]

McClelland, J. L. (1989). Parallel distributed processing and role assignment constraints. In Y. Wilks (Ed.), *Theoretical issues in natural language processing* (pp. 64–72). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 22]

McClelland, J. L., & Kawamoto, A. H. (1986). Mechanisms of sentence processing: Assigning roles to constituents of sentences. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models* (pp. 272–325). Cambridge, MA: MIT Press. [pg. 22–23]

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, *88*(5), 375-407. [pg. 18]

McClelland, J. L., & St. John, M. (1987). *Three short papers on language and connectionism* (Tech. Rep. No. AIP-1). Pittsburgh, PA: Department of Psychology, Carnegie Mellon University. [pg. 22]

McClelland, J. L., St. John, M., & Taraban, R. (1989). Sentence comprehension: A parallel distributed processing approach. *Language and Cognitive Processes*, *4*, 287–335. [pp. 17,20,22]

McClelland, J. L., & Thomas, A. (1996). *Remediation of language processing deficits: A Hebbian model.* Unpublished manuscript. [pg. 295]

McRae, K., Ferretti, T. R., & Amyote, L. (1997). Thematic roles as verb-specific concepts. *Language and Cognitive Processes*, *12*, 137–176. [pp. 47,50,192]

McRae, K., Spivey-Knowlton, M. J., & Tanenhaus, M. K. (1998). Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, *38*, 283–312. [pg. 50]

Miikkulainen, R. (1990a). Script recognition and hierarchical feature maps. *Connection Science*, *2*, 83–101. [pg. 24]

Miikkulainen, R. (1990b). A PDP architecture for processing sentences with relative clauses. In *COLING-90: Papers presented to the 13th International Conference on Computational Linguistics* (pp. 3/201–206). Helsinki. [pp. 24,28]

Miikkulainen, R., & Dyer, M. (1990). *Natural language processing with modular neural networks and distributed lexicon* (Tech. Rep. No. CSD-900001). Los Angeles, CA: Computer Science Department, University of California. [pg. 23]

Miikkulainen, R., & Dyer, M. G. (1989a). Encoding input/output representations in connectionist cognitive systems. In D. Touretzky, G. Hinton, & T. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School* (pp. 347–356). Los Altos, CA: Morgan Kaufman. [pg. 23]

Miikkulainen, R., & Dyer, M. G. (1989b). A modular neural network architecture for sequential paraphrasing of script-based stories. In *Proceedings of the International Joint Conference on Artificial Intelligence, IEEE* (pp. II/49–56). [pg. 23]

Miikkulainen, R., & Dyer, M. G. (1991). Natural language processing with modular PDP networks and distributed lexicon. *Cognitive Science*, *15*, 343–399. [pp. 23,28]

Miller, G. A., & Chomsky, N. (1963). Finitary models of language users. In R. D. Luce, R. R. Bush, & E. Galanter (Eds.), *Handbook of mathematical psychology, vol. 2* (pp. 419–491). New York: John Wiley and Sons, Inc. [pg. 7]

Miller, G. A., & Isard, S. (1964). Free recall of self-embedded English sentences. *Information and Control*, *7*, 202–303. [pp. 33,38,41–42]

Mitchell, D. C. (1987). Lexical guidance in human parsing: Locus and processing characteristics. In M. Coltheart (Ed.), *Attention and performance xii: The psychology of reading* (pp. 601–618). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 57,219,224]

Mitchell, D. C. (1994). Sentence parsing. In M. A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 375–409). Academic Press. [pg. 57]

Mitchell, D. C., Corley, M. M. B., & Garnham, A. (1992). Effects of context in human sentence parsing: Evidence against a discourse-based proposal mechanism. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *18*, 69–88. [pp. 63,65]

Mitchell, D. C., & Holmes, V. M. (1985). The role of specific information about the verb in parsing sentences with local structural ambiguity. *Journal of Memory and Language*, *24*, 542–559. [pp. 52–53,56–57,205,219,224]

Morgan, J. L., Bonamo, K. M., & Travis, L. L. (1995). Negative evidence on negative evidence. *Developmental Psychology*, *31*, 180–197. [pg. 8]

Morgan, J. L., & Travis, L. L. (1989). Limits on negative information in language input. *Journal of Child Language*, *16*, 531–552. [pg. 8]

Nakagawa, H., & Mori, T. (1988). A parser based on connectionist model. In *COLING Budapest: Proceedings of the 12th International Conference on Computational Linguistics* (pp. 454–458). Budapest. [pg. 19]

Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive Science*, *34*, 11–28. [pg. 25]

Ni, W., & Crain, S. (1990). How to resolve structural ambiguities. In *Proceedings of the 20th annual meeting of the North Eastern Linguistic Society* (pp. 414–427). Amherst, MA: Graduate Linguistics Students Association. [pp. 47,50,63,96–97]

Noelle, D. C., & Cottrell, G. W. (1995). A connectionist model of instruction following. In *Proceedings of the 17th annual conference of the Cognitive Science Society* (pp. 369–374). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 23]

O'Seaghdha, P. G., Dell, G. S., Peterson, R. R., & Juliano, C. (1992). Models of form-related priming in comprehension and production. In R. G. Reilly & N. E. Sharkey (Eds.), *Connectionist approaches to natural language processing* (pp. 373–408). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 27]

Pearlmutter, B. A. (1989). Learning state space trajectories in recurrent neural networks. *Neural Computation*, *1*, 263–269. [pp. 21,127]

Pearlmutter, N. J., & MacDonald, M. C. (1992). Plausibility and syntactic ambiguity resolution. In *Proceedings of the 14th annual conference of the Cognitive Science Society* (pp. 498–503). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 45–48,50,191–192,202]

Pearlmutter, N. J., & MacDonald, M. C. (1995). Individual differences and probabilistic constraints in syntactic ambiguity resolution. *Journal of Memory and Language*, *34*, 521–542. [pg. 49–50]

Pearlmutter, N. J., & Mendelsohn, A. A. (1999). *Serial versus parallel sentence comprehension.* Unpublished manuscript. [pg. 5]

Pickering, M. J., & Branigan, H. P. (1998). The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and Language*, *39*, 633–651. [pg. 68]

Pickering, M. J., & Traxler, M. J. (1998). Plausibility and recovery from garden paths: An eye-tracking study. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *24*, 940–961. [pp. 219,224]

Pickering, M. J., Traxler, M. J., & Crocker, M. W. (2000). Ambiguity resolution in sentence processing: Evidence against frequency-based accounts. *Journal of Memory and Language*, *43*, 447–475. [pp. 55–56,58,208–209,214,217,220,224,282]

Pineda, F. J. (1989). Recurrent backpropagation and the dynamical approach to adaptive neural computation. *Neural Computation*, *1*, 161–172. [pg. 127]

Pinker, S. (1984). *Language learnability and language development.* Cambridge, MA: Harvard University Press. [pg. 8]

Pinker, S. (1994). *The language instinct.* New York: Morrow. [pg. 7]

Plaut, D. C., & Kello, C. T. (1999). The emergence of phonology from the interplay of speech comprehension and production: A distributed connectionist approach. In B. MacWhinney (Ed.), *The emergence of language* (pp. 381–415). Mahwah, NJ: Lawrence Erlbaum Associates. [pg. 251]

Plaut, D. C., & Shallice, T. (1993). Deep dyslexia: A case study of connectionist neuropsychology. *Cognitive Neuropsychology*, *10*, 377–500. [pg. 128]

Pollack, J. (1988). Recursive auto-associative memory: Devising compositional distributed representations. In *Proceedings of the 10th annual conference of the Cognitive Science Society* (pp. 33–39). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 29,118]

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, *46*, 77–105. [pp. 21,29,118]

Rager, J. E. (1992). Self-correcting connectionist parsing. In R. G. Reilly & N. E. Sharkey (Eds.), *Connectionist approaches to natural language processing* (pp. 143–167). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 19]

Rayner, K., Carlson, M., & Frazier, L. (1983). The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences. *Journal of Verbal Learning and Verbal Behavior*, *22*, 358–374. [pp. 45,59–61,181,190,192]

Rayner, K., & Frazier, L. (1987). Parsing temporarily ambiguous complements. *Quarterly Journal of Experimental Psychology*, *39A*, 657–673. [pp. 52,205]

Rayner, K., Garrod, S., & Perfetti, C. A. (1992). Discourse influences during parsing are delayed. *Cognition*, *45*, 109–139. [pp. 60,63–66]

Reich, P. A. (1969). The finiteness of natural language. *Language*, *45*, 831–843. [pg. 4]

Rohde, D. L. T., & Plaut, D. C. (1997). Simple recurrent networks and natural language: How important is starting small? In *Proceedings of the 19th annual conference of the Cognitive Science Society* (pp. 656–661). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 25,142]

Rohde, D. L. T., & Plaut, D. C. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, *72*(1), 67–109. [pp. 9,12,25,128,140,142,284,287]

Rohde, D. L. T., & Plaut, D. C. (in press). Less is less in language acquisition. In P. Quinlan (Ed.), *Studies in developmental psychology: Connectionist models of development.* Charles Hume. [pp. 9,25,142]

Rumelhart, D. E., Durbin, R., Golden, R., & Chauvin, Y. (1995). Backpropagation: The basic theory. In Y. Chauvin & D. Rumelhart (Eds.), *Back-propagation: Theory, architectures, and applications* (pp. 1–34). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 10]

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations* (pp. 318–362). Cambridge, MA: MIT Press. [pp. 10,26,127–128]

Santorini, B. (1990). *Part-of-speech tagging guidelines for the Penn Treebank Project* (Tech. Rep. No. MS-CIS-90-47). Philadephia, PA: Department of Computer and Information Science, University of Pennsylvania. [pg. 76]

Schlesinger, I. M. (1968). *Sentence structure and the reading process.* The Hague: Mouton. [pp. 39,41–42]

Schütze, C. T. (1995). PP attachment and argumenthood. In C. T. Schütze, J. B. Ganger, & K. Broihier (Eds.), *Papers on language processing and acquisition, MIT working papers in linguistics, volume 26* (pp. 95–151). [pg. 60]

Schütze, C. T., & Gibson, E. (1999). Argumenthood and English prepositional phrase attachment. *Journal of Memory and Language*, *40*, 409–431. [pp. 59–62,181]

Sedivy, J., & Spivey-Knowlton, M. (1994). The use of structural, lexical and pragmatic information in parsing argument ambiguities. In J. C. Clifton, L. Frazier, & K. Rayner (Eds.), *Perspectives on sentence processing* (pp. 389–413). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 60–61,96]

Seidenberg, M. S., & MacDonald, M. C. (1999). A probabilistic constraints approach to language acquisition and processing. In M. H. Christiansen, N. Chater, & M. S. Seidenberg (Eds.), *Special Issue of Cognitive Science: Connectionist Models of Human Language Processing: Progress and Prospects* (Vol. 23). Cognitive Science. [pg. 17]

Selman, B., & Hirst, G. (1985). Connectionist parsing. In *Proceedings of the 7th annual conference of the Cognitive Science Society* (pp. 212–221). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 19]

Selman, B., & Hirst, G. (1994). Parsing as an energy minimization problem. In G. Adriaens & U. Hahn (Eds.), *Parallel natural language processing* (pp. 238–254). Norwood, NJ: Ablex Publishing. [pg. 19]

Sharkey, N. E., & Reilly, R. G. (1992). Connectionist natural language processing. In R. G. Reilly & N. E. Sharkey (Eds.), *Connectionist approaches to natural language processing* (pp. 1–12). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 17]

Small, S., Cottrell, G., & Shastri, L. (1982). Toward connectionist parsing. In *Proceedings of the National Conference on Artificial Intelligence* (pp. 247–250). Pittsburgh, PA: AAAI. [pg. 18]

Spivey, M., & Tanenhaus, M. (1998). Syntactic ambiguity resolution in discourse: Modeling the effects of referential context and lexical frequency. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *24*, 1521–1543. [pp. 146,284,291]

Spivey-Knowlton, M., & Sedivy, J. C. (1995). Resolving attachment ambiguities with multiple constraints. *Cognition*, *55*, 227–267. [pg. 60]

Spivey-Knowlton, M., & Tanenhaus, M. (1994). Referential context and syntactic ambiguity resolution. In J. C. Clifton, L. Frazier, & K. Rayner (Eds.), *Perspectives on sentence processing* (pp. 415–439). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 63–65]

Spivey-Knowlton, M., Trueswell, J. C., & Tanenhaus, M. (1993). Context effects and syntactic ambiguity resolution: Discourse and semantic influences in parsing reduced relative clauses. *Canadian Journal of Experimental Psychology*, *47*, 276–309. [pp. 63,65]

Steedman, M. (1999). Connectionist sentence processing in perspective. In M. H. Christiansen, N. Chater, & M. S. Seidenberg (Eds.), *Special Issue of Cognitive Science: Connectionist Models of Human Language Processing: Progress and Prospects* (Vol. 23). Cognitive Science. [pg. 17]

Stemberger, J. P. (1982). Syntactic errors in speech. *Journal of Psycholinguistic Research*, *11*, 313–345. [pg. 66–67]

Stevenson, S. (1994). *A competitive attachment model for resolving syntactic ambiguities in natural language parsing.* Unpublished doctoral dissertation, Department of Computer Science, University of Maryland. [pg. 20]

Stevenson, S., & Merlo, P. (1997). Lexical structure and parsing complexity. *Language and Cognitive Processes*, *12*, 349–399. [pg. 20]

St. John, M. F. (1992a). Learning language in the service of a task. In *Proceedings of the 14th annual conference of the Cognitive Science Society* (pp. 271–276). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 23]

St. John, M. F. (1992b). The story gestalt: A model of knowledge-intensive processes in text comprehension. *Cognitive Science*, *16*, 271–306. [pg. 24]

St. John, M. F., & McClelland, J. L. (1988). Applying contextual constraints in sentence comprehension. In *Proceedings of the 10th annual conference of the Cognitive Science Society* (pp. 26–32). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 21–23,117,131,133–134,138,286]

St. John, M. F., & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, *46*, 217–457. [pp. i,21–24,131]

St. John, M. F., & McClelland, J. L. (1992). Parallel constraint satisfaction as a comprehension mechanism. In R. G. Reilly & N. E. Sharkey (Eds.), *Connectionist approaches to natural language processing* (pp. 97–136). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 21–22,131,133]

Stolz, W. S. (1967). A study of the ability to decode grammatically novel sentences. *Journal of Verbal Learning and Verbal Behavior*, *6*, 867–873. [pp. 7,42]

Stowe, L. (1989). Thematic structures and sentence comprehension. In G. N. Carlson & M. K. Tanenhaus (Eds.), *Linguistic structure in language processing.* Dordrecht: Kluwer Academic Publishers. [pg. 234]

Sturt, P., Pickering, M. J., & Crocker, M. W. (1999). Structural change and reanalysis difficulty in language comprehension. *Journal of Memory and Language*, *40*, 136–150. [pg. 57–58]

Tabor, W. (1998a). *Context free grammar representation in neural networks.* Paper presented at the Neural Information Processing Systems (NIPS) workshop on Hybrid Neural Symbolic Systems. [pg. 294]

Tabor, W. (1998b). *Dynamical automata* (Tech. Rep. No. TR98-1694). Ithaca, NY: Computer Science Department, Cornell University. [pg. 294]

Tabor, W., Juliano, C., & Tanenhaus, M. (1996). A dynamical system for language processing. In *Proceedings of the 18th annual conference of the Cognitive Science Society* (pp. 690–695). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 144]

Tabor, W., Juliano, C., & Tanenhaus, M. K. (1997). Parsing in a dynamical system: An attractor-based account of the interaction of lexical and structural constraints in sentence processing. *Language and Cognitive Processes*, *12*(2/3), 211–271. [pp. 26,144,284]

Tabor, W., & Tanenhaus, M. K. (1999). Dynamical models of sentence processing. In M. H. Christiansen, N. Chater, & M. S. Seidenberg (Eds.), *Special Issue of Cognitive Science: Connectionist Models of Human Language Processing: Progress and Prospects* (Vol. 23). Cognitive Science. [pg. 144]

Tabossi, P., Spivey-Knowlton, M. J., McRae, K., & Tanenhaus, M. K. (1994). Semantic effects on syntactic ambiguity resolution: Evidence for a constraint-based resolution process. In C. Umilta & M. Moscovitch (Eds.), *Attention and performance xv* (pp. 589–615). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 46–47,50–51,152,192]

Taraban, R., & McClelland, J. L. (1988). Constituent attachment and thematic role assignment in sentence processing: Influences of content-based expectations. *Journal of Memory and Language*, *27*, 597–632. [pp. 6,59–61]

Trueswell, J. C. (1996). The role of lexical frequency in syntactic ambiguity resolution. *Journal of Memory and Language*, *35*, 566–585. [pp. 48,50–51,202]

Trueswell, J. C., & Tanenhaus, M. K. (1991). Tense, temporal context and syntactic ambiguity resolution. *Language and Cognitive Processes*, *6*, 303–338. [pp. 62–63,65]

Trueswell, J. C., & Tanenhaus, M. K. (1992). Consulting temporal context during sentence comprehension: Evidence from the monitoring of eye movements in reading. In *Proceedings of the 14th annual conference of the Cognitive Science Society* (pp. 492–497). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 63,65]

Trueswell, J. C., & Tanenhaus, M. K. (1994). Toward a lexicalist framework for constraint-based syntactic ambiguity resolution. In J. C. Clifton, L. Frazier, & K. Rayner (Eds.), *Perspectives on sentence processing* (pp. 155–179). Hillsdale, NJ: Lawrence Erlbaum Associates. [pp. 46,62,152,284]

Trueswell, J. C., Tanenhaus, M. K., & Garnsey, S. M. (1994). Semantic influences on parsing: Use of thematic role information in syntactic ambiguity resolution. *Journal of Memory and Language*, *33*, 285–318. [pp. 46,48,50–51,191–192]

Trueswell, J. C., Tanenhaus, M. K., & Kello, C. (1993). Verb-specific constraints in sentence processing: Separating effects of lexical preference from garden-paths. *Journal of Experimental Psychology*, *19*(3), 528–553. [pp. 53–56,207–209,213–214,217]

Waltz, D. L., & Pollack, J. B. (1985). Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, *9*, 51–74. [pg. 18–19]

Ward, N. (1991). *A flexible, parallel model of natural language generation.* Unpublished doctoral dissertation, Computer Science Division, University of California, Berkeley, CA. (UCB/CSD 91/629) [pg. 27]

Warren, P., Nolan, F., Grabe, E., & Holst, T. (1995). Post-lexical and prosodic phonological processing. *Language and Cognitive Processes*, *10*, 411–417. [pg. 229]

Weber, V., & Wermter, S. (1996). Using hybrid connectionist learning for speech/language analysis. In S. Wermter, E. Riloff, & G. Scheler (Eds.), *Lecture notes in artificial intelligence 1040: Connectionist, statistical, and symbolic approaches to learning for natural language processing* (pp. 87–101). Berlin: Springer-Verlag. [pg. 20]

Weckerly, J., & Elman, J. L. (1992). A PDP approach to processing center-embedded sentences. In *Proceedings of the 14th annual conference of the Cognitive Science Society* (pp. 414–419). Hillsdale, NJ: Lawrence Erlbaum Associates. [pg. 26]

Wermter, S., Riloff, E., & Scheler, G. (1996). Learning approaches for natural language processing. In S. Wermter, E. Riloff, & G. Scheler (Eds.), *Lecture notes in artificial intelligence 1040: Connectionist, statistical, and symbolic approaches to learning for natural language processing* (pp. 1–16). Berlin: Springer-Verlag. [pg. 17]

Wermter, S., & Weber, V. (1994). Learning fault-tolerant speech parsing with screen. In *Proceedings of the 12th National Conference on Artificial Intelligence* (pp. 670–675). Seattle, WA: AAAI. [pg. 20]

Wermter, S., & Weber, V. (1997). SCREEN: Learning a flat syntactic and semantic spoken language analysis using artificial neural networks. *Journal of Artificial Intelligence Research*, *6*, 35–85. [pg. 20]

Yngve, V. H. (1960). A model and an hypothesis for language structure. *Proceedings of the American Philosophical Society*, *104*, 444–466. [pg. 7]

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n^3. *Information and Control*, *10*(2), 189–208. [pg. 18]