The Role of Background Knowledge in Sentence Processing

Raluca Budiu

August, 2001

CMU-CS-01-148

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:
John R. Anderson, Chair
Jaime Carbonell
David Plaut
Lynne Reder, Department of Psychology

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA) under the Office of Naval Research (ONR) contract no. N020149910097 and the National Science Foundation under various grants. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of DARPA, ONR, NSF, the U.S. government or any other entity.

Keywords: sentence comprehension, computational model, metaphor, Moses illusion, sentence memory, ACT-R, scalability

Abstract

In this dissertation I describe a cognitive model of sentence processing. The model operates at the semantic level and can apply to verification or comprehension of metaphoric or literal sentences, isolated or embedded in discourse. It uses an incremental search–and–match mechanism to find a long-term–memory referent (interpretation) for an input sentence. The search is guided by cues such as the last few words read or previous tentative interpretations. The process of comprehension produces a propositional representation for the input sentence and also keeps track of local comprehension failures.

The model is implemented in the ACT-R framework and offers a scalable solution to the problem of language comprehension: its performance (in terms of speed and accuracy) is roughly invariant to the number of facts held in the long-term memory. Its predictions match data from psycholinguistic studies with human subjects. Specifically, the sentence-processing model can simulate the comprehension and verification of metaphoric and literal sentences, metaphor-position effects on sentence comprehension, semantic illusions and their dependence on semantic similarity between the distortion and the undistorted term. The products of the sentence-processing model can explain the pattern of sentence recall in text-memory experiments.

This dissertation also explores the modeling alternatives faced by the design of a sentence-processing model. I show that, to achieve comprehension speed comparable to that of humans, a model must minimize the explicit search process and rely on semantic associations among words. I also investigate how the representation chosen for propositions and meanings affects the comprehension process in a production-system framework such as ACT-R.

Acknowledgments

I am deeply grateful to my advisor, John Anderson, who constantly encouraged my interest in psycholinguistics and metaphor. His countless ideas and engaging arguments contributed substantially to this dissertation. He closely guided my work and helped me monitor my goals. His passion for research and his energy have always inspired me.

I thank my committee members, Jaime Carbonell, David Plaut, and Lynne Reder. They offered me valuable and diverse perspectives on language comprehension and encouraged me to pursue different approaches to the subject. I am grateful to Lynne Reder for her trust and appreciation of my work and for always trying to raise my confidence level. I thank her for graciously supplying me with her Moses-illusion data.

I am indebted to Mike Ayers for kindly providing me his manuscript and data on Moses illusion. I thank Jose Quesada for collecting LSA similarities for me.

I also thank my friends who made my stay in Pittsburgh a highly pleasurable experience: Cristian Dima, Nathalie Evrat, Marius Minea, George Necula, Dan Qu, Ion Stoica. Marius was a endless source of useful information. I cherish the long and challenging contradictory discussions on art, movies, and literature that I had with Dan. Ion's inquisitive and passionate nature always brought a fresh perspective in our conversations.

I am profoundly grateful to Mihai Budiu, for his continual support and for his effervescent enthusiasm. He had the patience to listen to endless perseverations related or not to my research; he gave me helpful questions and advice (some of it included in this thesis) and he took time from his duties to read and comment on this dissertation. Mostly, he made me laugh every day.

Contents

List of Figures

List of Tables

Chapter 1

Overview and Contributions

Language is a basic experience for most of us: we use it everyday for communicating with others or for expressing ourselves. The majority of children acquire their parents' language relatively easily. Nonetheless, understanding the nature of language is no trivial task: computer programs that attempt natural-language comprehension have only limited capabilities. One feature of human language that is often frustrating for computer programs is ambiguity: not only can words have multiple meanings, but sometimes the meaning of a word depends on the cultural knowledge of the two interlocutors. Think, for instance, at the meaning of the phrase *Bless you*: if it is said to a person who has just sneezed, it is interpreted differently than if the speaker is a priest; moreover, in certain cultures it is considered impolite to say *Bless you* unless you are familiar to the person who sneezed. The choice of words or language registers adds meaning to the utterance.

Even if we disregard the broader issue of cultural knowledge, words are not always taken at face value. In the indirect request *Could you pass me the salt?*, the word *could* is not used literally: the speaker is usually not interested in the ability of his dialogue partner. Everyday language is often nonliteral; figurative devices such as irony (e.g., "What lovely weather we're having" stated in the midst of a rainstorm), metaphor (e.g., "his marriage is an icebox"), metonimy (e.g., "The ham sandwich asked for the check", uttered by a restaurant waiter), or hyperbole (e.g., "I have a thousand papers to read until tomorrow") are common and are understood quite easily. Metaphor is a particularly interesting such device: it shapes and enriches the language. A lot of words in contemporary language were initially metaphors: the word *tortoise* comes from the Old Greek *tartarouchos*, meaning "of Tartarus", which to the Greeks signified the infernal regions; the word for *language* is in many languages the same as the word for *tongue*, the organ of speech; and, since Internet became ubiquitous, the word "Web" has a new meaning. Not only do metaphors contribute to language change, but, according to researchers such as Lakoff (Lakoff, 1987; Lakoff & Johnson, 1990) or Reddy (1993), language is often shaped by existing, conceptual metaphors —- for example, the words we use for the concepts related to communicating ideas are generated by the "language-as-a-conduit" metaphor — *give somebody a good idea*, *take somebody's idea*, *pack thoughts into words*, *his words are hollow*, and so on.

The human language-comprehension mechanism succeeds not only in the presence of ambiguity or of nonliterality, but also of noise. Real-time communication is inherently noisy — often, the communication medium is suboptimal (think of a bad connection on the phone or of a discussion in a noisy room) and people make errors in pronunciation or choice of words, but their communication partners are able to grasp the gist of their message. Sometimes speakers say what they did not intend, but we still understand them (for example, think of the *lapsus linguae*, slips of the tongue, made famous by Sigmund Freud). Not only do listeners often recover from mispronunciations or slips of the tongue, but sometimes they are unable to notice them in a sentence. For instance, when asked *When an aircraft crashes, where should the survivors be buried?* about 80 percent of people do not detect the anomaly (i.e., that the survivors need not be buried) (Barton & Sanford, 1993). Even if they are warned in advance that the sentence may be distorted, about 40 percent of subjects still do not notice the inconsistency in a statement such as *Moses took two animals of each kind on the ark*, in spite of knowing that *Noah*, rather than Moses, is the character of the ark story in the Bible (Erickson & Mattson, 1981). (This phenomenon is called **Moses illusion**.) These facts seem to suggest that ignoring minor discrepancies in communication is such a basic feature of our language system that we cannot easily turn it off. An insensitivity to minor slips certainly helps at making communication reliable, but does it serve any other purpose?

In this dissertation I argue that metaphor comprehension and lapses in detecting semantic inconsistencies are facets of the same mechanism of language processing; that we understand metaphors easily for the same reasons for which we fail to notice semantic distortions. Specifically, I propose a theory of sentence understanding that accounts for how people comprehend everyday-language metaphors and ignore errors made by their communication partners. Intuitively, this theory claims that, if the sentence context is rich and supportive, it can help people grasp "instantaneously" the intended meaning of a metaphor or of an illusion, without going through the burden of detecting a literal mismatch between a metaphoric or distorted word and the other words in a sentence. That is, the literal meaning of a word can be bypassed if the other words in the sentence are informative enough.

The theory is embodied in an ACT-R (Anderson & Lebiere, 1998) model. ACT-R is a cognitive architecture that served as a framework for successfully modeling a large variety of problem-solving and memory tasks (see `http://act.psy.cmu.edu/papers/ACT-R_Models.htm` for a list of articles that describe ACT-R models). At a very superficial level, ACT-R is a programming language; however, the constructs in this language reflect assumptions about the human cognition. Implementing the sentence-processing theory in ACT-R has a number of advantages. First, unlike human language, which is inherently ambiguous, the success of programming languages relies on their lack of ambiguity. Thus, being a rigorous programming-language, ACT-R eliminates the ambiguity in the description of a theory and enables a rigorous articulation. Second, ACT-R makes specific latency predictions for the actions performed by a model. Thus, the output produced (e.g., response) and the time taken by an ACT-R model for a task can be compared directly to the output and time produced by human subjects on the same task. Third, ACT-R models bear with them sophisticated architectural assumptions about human cognition that were repeatedly tested in other models.

This dissertation describes the ACT-R sentence-processing model and evaluates it based on several criteria. One test of this model is empirical: the model-generated responses and response latencies are compared with behavioral data obtained from human subjects. A second criterion is computational: it assesses whether the model is scalable or correct (in the sense that it produces the same response as people do). Last but not least, I evaluate the model by contrasting it with others existent in the field. The rest of this chapter offers a bird's-eye view of the computational model that I propose and discusses the evaluation criteria.

1.1 Main Contribution

In this dissertation I describe a model of how people comprehend sentences. This model can equally understand literal, metaphoric, or semantically distorted sentences, isolated or embedded in discourse. Figure 1.1 shows its basic inputs and outputs: the model receives the words one by one (as we do in speech or reading) and, based on some background knowledge about the world, it finds an **interpretation** (or a meaning) for the sentence formed by the input words. The background knowledge contains information resulting from past experience or from preceding discourse. The interpretation is a proposition in the background knowledge that best matches the input sentence. Thus, the model defines comprehension of a sentence as finding a proposition in the background knowledge that matches that sentence.

The search for an interpretation is **on line**: after each word "read", the model eagerly attempts to "guess" the meaning of the entire utterance. In its pursuit for finding an interpretation, the model uses cues such as the last few words read or previous candidate interpretations, as well as information about the syntactic structure of the input sentence; such cues dramatically reduce the search space.

Starting from these simple assumptions, there are three aspects that describe the sentence-processing model:

1. How the interpretation is **selected** from the background knowledge

2. When a proposition is considered to **match** the input sentence

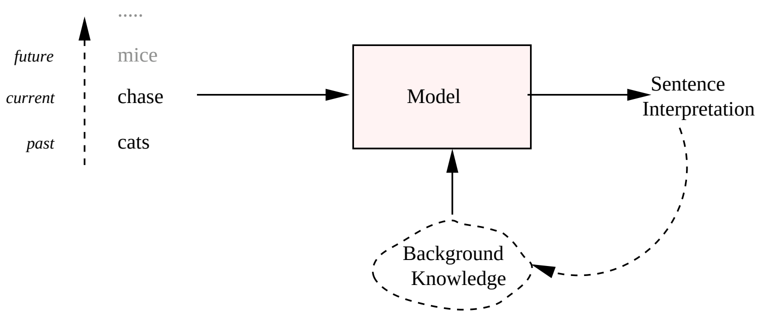


Figure 1.1: The inputs and output of the sentence comprehension model. The input words correspond to the sentence *Cats chase mice.*

3. How (and whether) the candidate interpretation found influences comprehension of the next words (I sometimes refer to this aspect as **interpretation priming**)

Whereas it is hard to dispute that language processing involves prior knowledge, it may seem strange that comprehension is defined as finding a matching proposition in background knowledge. Does this definition mean that we cannot understand sentences that transmit new information, with no correspondent in background knowledge? The answer to this question depends on how flexible the definition of "matching" is. In this model, I assume that understanding of novel facts is done by relating them to similar, more familiar propositions, based on the information that the new facts share with older ones. Thus, a sentence may match a background-knowledge proposition if it simply evokes that proposition, in spite of not containing the same information.

1.2 Other Contributions

Beside building one model that captures a number of different human behaviors, this dissertation explores some of the options available for modeling sentence processing. For each of the three defining aspects of the model that were identified in the previous section (i.e., search, matching, interpretation priming), there are several dimensions on which the modeling can vary — for instance, the search for an interpretation may be more or less informed, the criteria for matching of a proposition to a sentence may vary in how lax they are, a candidate interpretation, even if proved wrong, may contribute or not to the search for new interpretations. Another interesting set of choices is related to the appropriate representation for propositions and meanings. For instance, the meaning of a word could be regarded as a set of semantic features; in that case, processing one word would mean processing (some of) its features. Alternatively, the word meaning could be atomic, encapsulated into one unit. The same atomic–distributed distinction can be made at the level of propositions:

the concepts occurring in one proposition could be either grouped together in memory[1] or could be represented separately as distinct parts of the proposition.

Yet another domain that this thesis explores is the relationship of such a sentence-processing model with a complex cognitive architecture such as ACT-R. Aside from the high-level modeling choices that were mentioned earlier, there were a number of ACT-R choices that one had to make. Was there a better way of using ACT-R capabilities to do sentence processing? Were any of the ACT-R features vital for this model? I address such questions in this dissertation.

1.3 Limitations

Whereas my computational model is able to perform well on various sentence-processing tasks, it is based on several simplifying assumptions. Its most obvious limitation is that it does not account for syntactic processing. All its computations are semantic; the model does not create any parse tree for the input sentence and does not assign thematic roles (i.e., agent, patient, oblique, etc.) to words. Rather, it assumes that the thematic role for each word is known in advance and uses that information to find an interpretation. Nonetheless, the correct thematic role of a word need not be provided to the model. Were the thematic role proven wrong, the model would still be able to recover and, in many cases, to find an interpretation for the input sentences. However, this dissertation does not deal with such cases explicitly.

In Section 1.1 we saw that the sentence-processing model attempts to find an interpretation on line, as it reads each word. However, the granularity is not strictly at the word level, but rather at the noun-phrase or verb-phrase level. Thus, the modifiers of a noun or of a verb are considered together with the noun, respectively verb, rather than as separate words. For instance, the noun phrase *blue drops of ink* is analyzed as one word and its meaning is considered part of the model's lexicon. Whereas there is behavioral evidence that basic aspects of sentence comprehension are performed on line (Marslen-Wilson, 1975, 1973), it is possible that incrementality[2] is at the level of groups of words rather than individual words — that is, rather than searching for an interpretation after each word, people may do so at every two or three words. However, even if the noun- or verb-phrase granularity were supported by empirical data, my model would still have to solve the problem of computing the meaning of the noun (or verb) phrase out of its components.

In the next section, we see that one way to evaluate the model is by comparing its behavior with that of human subjects on the same task. Yet, as the syntactic processes are not modeled, the latencies produced by the model should be taken *cum grano salis*. Whereas the relative trends in latency among various conditions are captured because of intrinsic

[1] I use the terms "background knowledge" and "(long-term) memory" interchangeably.

[2] I use the words "incremental" and "on line" as synonyms.

properties of the model, getting identical latencies from model and from people should not be considered as the main accomplishment of the model, given that the model performs a simpler task. The model attempts to compensate for such differences by assigning extra processing costs per word, but perhaps other factors in the syntax of the sentence could affect the processing time.

Another limitation is related to semantic similarities between words. In many simulations, I assume there is a similarity between the two terms of a metaphor (e.g., between *time* and money for the metaphor *time is money*) or between different distortions (e.g., *Moses* and *Noah*) and the undistorted terms in a Moses-illusion task. Although the setting of such similarities is qualitatively reasonable and sometimes supported by experimental data or by theories of metaphor, nowhere are they based on actual similarity ratings collected from people.

My theory addresses sentence comprehension and verification in context. However, it is not a full-fledged theory of how people process sentences embedded in text: it does not deal with drawing causal inferences, resolving pronominal references, or even binding multiple references to the same object. Although the model produces a discourse representation that reflects a number of relationships between various text propositions, that representation is by no means complete or coherent. In developing this theory, I focused on sentence processing rather than on discourse processing; the extension of the theory to text input was mainly in the attempt to explain comprehension metaphors embedded in larger text.

1.4 Evaluation

This dissertation describes three methods of evaluation for the sentence-processing model: empirical evaluation, computational evaluation, and comparison with other models in the literature. Because the objective is to produce a model of human comprehension, it is natural to compare the performance of the model with that of humans, either on specific tasks devised in the laboratory (evaluation!empirical) or on simpler tasks that we take for granted in humans (evaluation!computational). The empirical criterion of evaluation consists in using the model for simulating human data from a variety of psychology experiments. The computational criteria refer to whether the model comprehends (simple) literal sentences fast enough, whether it produces correct interpretations for them, and whether its speed and accuracy depend on the number of facts in the background knowledge.

1.4.1 Empirical Evaluation

I test the sentence-processing model on three types of data: metaphoric sentences, Moses-illusion sentences, and sentence memory. The latter may seem a strange test for a model of comprehension; however, I show that the connections established at comprehension between a sentence and its background-knowledge interpretation play an active role in the recall of the sentence.

Metaphor Comprehension and Verification. We saw that, in spite of the common belief that they are the appanage of writers, metaphors are often encountered in everyday language. Not only is metaphor widely used, but in certain contexts metaphor is very easily understood. Many psycholinguistic studies attempted to compare metaphor and literal comprehension. Some of these studies found that subjects read a metaphoric sentence as fast as a literal one and concluded that the same processes are involved in the comprehension of literals and metaphors. However, other studies indicated that subjects show some difficulty in understanding (at least some) metaphors. It is a challenge for a theory of sentence processing to explain both these types of evidence.

My sentence-processing model works for both metaphoric and literal sentences, suggesting that processing is very similar in both cases. Indeed, the model can process a metaphor as smoothly as a literal if the metaphor is familiar or good (i.e., if the two domains compared by the metaphor are similar) or if the sentence context is supportive. The relative importance of these factors varies with the position of the metaphor within the sentence. If neither the metaphor properties nor the sentence context help, then the model must do extra work to comprehend the metaphoric sentence.

Next we briefly present three metaphor studies simulated in this dissertation. The first study (Gerrig & Healy, 1983) is concerned with how the position of the metaphor within a sentence affects the difficulty of processing. The other two provide mixed evidence for the similarity of metaphoric and literal processing — they show that, whereas people may understand metaphoric sentences as fast as literal sentences (Budiu & Anderson, 2000a), they take longer to verify metaphoric paraphrases of studied sentences than literal equivalents, if the metaphor is unfamiliar(Budiu & Anderson, 2001). Budiu and Anderson's (2000a) study also looks at how people understand every word in a metaphoric sentence and, thus, it offers a closer view of the step-by-step comprehension processes.

Gerrig and Healy (1983) compared reading latencies for metaphor-first sentences (e.g., *Drops of molten silver filled the night sky*) and metaphor-last sentences (e.g., *The sky was filled with drops of molten silver*). They found that subjects read metaphor-last sentences faster than metaphor-first sentences. This experiment showed that sentence context can speed up the processing of a metaphor, if it precedes it.

Budiu and Anderson (2000a) examined how people understand sentences containing one or two metaphors. They looked at four types of sentences, each obtained by manipulating the metaphoricity of either the noun or the verb. Thus, after a passage about a women's meeting, participants would read *The hens/women clucked/talked noisily*[3]. Even though

[3] This notation is a shortcut for four sentences: *The hens clucked noisily, The hens talked noisily,* and two more variants.

globally there was no difference among the reading times of the four sentence types, the reading times for the individual words in the sentence differed for the metaphoric and literal conditions, with subjects taking more time to read the noun and the verb of sentences with metaphorical nouns, but being faster on their endings.

The third dataset selected for simulation belongs also to Budiu and Anderson (2001) and concerns verification of metaphoric and literal sentences. In that experiment, participants judged the truth of metaphoric and literal paraphrases of sentences from a passage. For instance, participants read a passage about an bulky athlete who was tired and slept in class and, then, had to verify the probe *The bear slept in class.* The same metaphors (preceded by different passages) were shown several times. At the beginning of the experiment, when the metaphors were new, people were slower and less accurate at judging metaphoric sentences than at judging literal sentences and they had a tendency to interpret metaphors literally. In contrast, by the end of the experiment, when the metaphors became familiar, participants comprehended metaphors and literals comparably fast. The data from Budiu and Anderson (2001) provides an interesting counterbalance to the Budiu and Anderson (2000a) study: although there may be no difference between metaphoric and literal targets in term of reading times, there seems to be one when people have to monitor their comprehension more closely (at least if the metaphors are less familiar).

Semantic Illusions. We saw that people often fail to notice distortions in sentences such as *How many animals of each kind did Moses take on the ark?* When confronted with such a question, subjects usually respond *two*, even if they are asked to monitor for distortions and even if they know that Noah, rather than Moses, took the animals on the ark (Erickson & Mattson, 1981).

Semantic illusions are a very robust and intriguing phenomenon: most manipulations intended to increase people's awareness of the illusion failed (Reder & Kusbit, 1991; Kamas, Reder, & Ayers, 1996). However, not all distortions are hard to notice. When, in the ark question, *Moses* was replaced with *Adam,* subjects were less likely to fall for the illusion (Ayers, Reder, & Anderson, 1996). Also, certain "illusions" simply don't work: nobody is tricked by *Who was the first man to walk on the sun?*

The Moses-illusion studies used two kinds of tasks: the "literal" task, in which subjects had to detect distortions, and the "gist" task, in which subjects had to answer the distorted questions as if they were undistorted (i.e., they should answer *two* when asked *How many animals of each kind did Moses take on the ark?*). Unlike for the "literal" Moses-illusion task, people are very good at the gist task (Reder & Kusbit, 1991; Ayers et al., 1996); they give correct responses faster in the gist task than in the literal task (Reder & Kusbit, 1991).

Moses-illusion phenomena indicate that comprehension processes are hard to control explicitly. They bear similarity to metaphor comprehension: people easily understand (at least some) metaphors and their difficulty in detecting the distortion may be connected to the ease with which they comprehend the message behind the distorted sentences. The gist task resembles metaphor comprehension even more than the literal task does: if in the literal task, subjects must look for errors, in the gist task the comprehension is unhindered by such monitoring and people must find a core meaning of the sentence, ignoring the literal sense of some words that are part of it.

Next I discuss in brief the Moses-illusion results chosen for being simulated by the sentence-processing model. They belong to two studies (Reder & Kusbit, 1991; Ayers et al., 1996) that compare (in terms of speed and accuracy) subjects' answers to distorted and undistorted questions, in both gist and literal tasks.

Reder and Kusbit (1991) reports several experiments attempting to identify the factors that influence subjects' awareness of the illusion. Two findings from that study are modeled in this dissertation — the first is that people give correct answers faster in the gist task than in the literal task; the second is the absence of a latency difference between judging correctly an undistorted sentence and a distorted one.

Ayers et al. (1996) looked at how often people fall for distortions. They showed subjects three variants of the same distorted question, one containing a good distortion (*How many animals of each kind did Moses take on the ark?*), another containing a bad distortion (*How many animals of each kind did Adam take on the ark?*), and the third being the undistorted question (*How many animals of each kind did Noah take on the ark?*). As expected, subjects fell for the good distortions more often than for the bad distortions. Similarly, in the gist task, they were able to answer the good-distortion questions correctly more often than they answered the bad-distortion questions. These findings suggest that, the more similar the distortion is to the undistorted term, the smoother the comprehension process and the harder the detection of inconsistencies. Ayers et al.'s (1996) results provide a nice counterpart to metaphor studies(Blasko & Connine, 1993) showing that metaphor goodness (which is possibly related to the similarity of the two domains compared by the metaphor) can play a role in the ease of metaphor comprehension.

Text Memory. Since Bartlett (1932) a lot of studies have shown the influence of prototypical knowledge (schemas and scripts — see Schank & Abelson, 1977) on text memory. For instance, Bransford and Johnson (1972) proved that people remember better a text if the text topic is explicitly mentioned before the text is read, as the readers can use their old knowledge about the subject to understand the text.

My model is not a model of sentence memory. Nonetheless, it does predict certain patterns of recall for studied text, based on the nature of the structures formed at study. Indeed, when the text is read, an interpretation is found for each sentence. If that interpretation is remembered later, its relationships with other propositions in the background knowledge can affect the recall. For instance, if the sentence *Joe paid the waiter* could be recalled, one could also infer the sentence *Joe ate the meal*, just because the interpretation of the first proposition is related to the interpretation of the second.

In this dissertation I simulate recall data from an experiment described by Bower, Black, and Turner (1979), who found that, if subjects study several stories on the same subject (e.g., a story about a visit to a doctor and another about a visit to a dentist), they tend to recall more facts not stated in the stories than if they study one story. The propositions incorrectly recalled are consistent with the subject of the stories (i.e., with the paradigmatic situation of visiting a health professional).

To summarize, one of the purposes of this thesis is to build a unique model that explains all these empirical phenomena. The intuition behind an unification of metaphor comprehension and semantic illusions is that in both cases the context "helps" with getting the gist of the sentence and, thus, compensates for the lack of literality or for the distortion. On the other hand, memory for text depends on the structures formed at study; those structures are products of comprehension. I argue that, whereas metaphor understanding and semantic illusions are both effects of the mechanism of comprehension, (at least some) text-memory phenomena can be explained based on the products of comprehension and that the long-term memory interpretation for the studied sentences helps or interferes with recognition or recall.

1.4.2 Computational Evaluation

A plausible sentence-processing model must behave as humans do in similar conditions. People typically understand literal sentences at a rate of about 500 ms per content word. They deal with relatively large knowledge bases and, even though under certain conditions such as time pressure or stress, they can confuse who did what, most of the time they can be trusted with correct comprehension.

We can identify a set of necessary computational constraints that a model of sentence processing must obey to be considered an acceptable theory. They include speed, scalability, and correctness.

Speed. As mentioned earlier, human comprehension is a relatively fast process. In a system in which a task is performed by firing a set of rules and each such firing takes a minimum amount of time, speed is a critical constraint: one simply cannot have too many rules firing for a single input sentence. The complexity of a sentence-processing model is thus severely limited. In a previous section we saw that my model searches for a matching proposition in the long term memory. Because the number of steps in the search cannot be very high, to produce reasonable results, the search must be quite informed (i.e., it cannot afford too many misses).

Scalability. People can acquire vast amounts of knowledge in a lifetime. It is reasonable to assume that their long term memories are very rich in information. However, this richness does not affect the time taken to understand a sentence. A model would not be acceptable if it took the right amount of time to understand a sentence when its background knowledge were on the order of a couple of propositions, but it took minutes to understand the same sentence if the background knowledge were closer to that of humans. Ideally, the comprehension time should not depend on the number of facts in long term memory.

Correctness. Most often communication between people is successful at the language level. If told *Mary ate the soup,* most English speakers would understand neither *John drank the soup* nor *The dog chased the cat.* One could imagine a model that would simply select an arbitrary fact from long term memory to be the meaning of the input sentence. For obvious reasons such a model would not be acceptable. Thus, a plausible model should (probabilistically) make errors where people are prone to make errors and be correct otherwise.

A subtle aspect of correctness is **role confusion**. Although people sometimes may mix agents with patients, they usually do it if the fillers of the confused roles are similar enough. It is rare for people to confuse roles in sentences such as *The lady lost the purse.*

Garden paths represent an instance of local role confusion that later is repaired; they refer to situations in which people assume one role assignment and then switch to another, when an inconsistency with the first is detected. Perhaps the most famous case of a garden path is *The horse raced past the barn fell:* most people who hear this sentence for the first time assume that *horse* is the agent of *race* and then realize that it is actually its patient.

1.4.3 Comparison with Other Models

One way of evaluating a new theory is by analyzing how it relates to other theories in the field. In this section I review very briefly some of the extant models of sentence processing; Chapter 10 includes a more detailed analysis and comparisons between these models and my model.

The most well-known theory of sentence processing is the construction–integration theory (Kintsch, 1988; Kintsch & van Dijk, 1978), which has been applied to a number of sentence- and discourse-comprehension tasks. As its name suggests, Kintsch's model has two phases: a symbolic-like construction phase in which the model builds a set *(text base)* of possible interpretations for an input and a connectionist-like integration phase, in which one of these interpretations is selected[4]. Knowledge is represented as an associative network, in which concepts and propositions are the nodes and the links have strengths reflecting (positive or negative) associations between them.

[4] I reformulated the CI model in terms specific to my model; according to Kintsch in the integration phase "the text base is integrated into a coherent whole".

Recently Kintsch (2000) proposed a way to incorporate comprehension of *A–is–B* metaphors into his CI theory, using assumptions and concepts from the latent semantic analysis (Landauer & Dumais, 1997). Latent semantic analysis (LSA) is a theory of meaning that represents words in relationship to the contexts in which they occur. Specifically, each meaning can be described as a vector in a semantic space; similarity between words is captured by the cosine of the angle formed by the vectors corresponding to the two words. For predicative metaphor understanding, Kintsch (2000) uses an associative network in which the connection strengths are given by LSA distances between corresponding concepts.

Some of the ideas present in my theory of metaphor understanding and verification stemmed from the classic given–new strategy proposed by Clark and Haviland (Clark, 1973a; Clark & Haviland, 1974; Haviland & Clark, 1974). According to that theory, normal sentences contain both old, *given* information and *new* information. The listener first "searches the memory for antecedent information that matches the sentence's given information; then he revises memory by attaching the new information to that antecedent" (Haviland & Clark, 1974).

My model of Moses illusion is, to some degree, similar to that proposed by Reder and her colleagues (Reder & Kusbit, 1991; Kamas & Reder, 1995; Kamas et al., 1996). After examining alternative possibilities, Reder and Kusbit (1991) concluded that the explanation most consistent with their data is partial matching between the critical concept (i.e., distortion) and the sentence context. Kamas and Reder (1995) and Kamas et al. (1996) elaborate the partial-matching hypothesis by assuming a spreading-activation mechanism in a semantic network: through that mechanism, "the more activation that accrues at a concept through its connection to the remainder of the concepts in the question, the more likely the person is to accept the retrieved concept as matching the question."

Although my model can understand sentences embedded in context, it is not a complete theory of discourse processing (for instance, it does not attempt to solve pronominal anaphoras or to bind various referents to the same concept). However, it is interesting to compare it with more refined models of discourse processing such as the memory-based text processing (MBP — see Lorch, 1998 for a review), if not necessarily in terms of problems solved then maybe in terms of ideas incorporated in its design. The basic idea of MBP (Myers & O'Brien, 1998; Noordman & Vonk, 1998; Cook, Halleran, & O'Brien, 1998; Albrecht & Myers, 1998) is that processes involved in text reading are derived from basic memory processes. Thus, as you read, the activation from the current focus spreads to related information that occurred previously in the text and makes that information available (this process is called *resonance*). Therefore, the reader need not make any explicit inferences to maintain local or global coherence; the only inferences are those due to past information "dumbly" evoked by the current words, regardless of its relevance. Although all MBP approaches claim that the same activation-spreading processes govern both the integration with the background knowledge and the episodic text memory, they mainly deal with the episodic text memory.

1.5 Thesis Organization

Chapter 2 presents a bird's-eye view of the basic mechanisms of ACT-R.4.0. Chapter 3 describes the basic model, which comprehends isolated sentences; in Chapter 4 I use that model to simulate Gerrig and Healy's (1983) metaphor-position experiment and two Moses-illusion experiments from studies by Reder and Kusbit (1991) and, respectively, by Ayers et al. (1996). Chapter 5 embellishes the basic model with mechanisms for sentence verification in context and for novel-sentence comprehension in context; this enriched model is tested against metaphor-comprehension data (Budiu & Anderson, 2001, 2000a) in Chapter 6 and against sentence-memory data (Bower et al., 1979) in Chapter 7. Chapter 8 examines the choices made in the design of the model and some alternatives to them. Chapter 9 evaluates the model in terms of the computational constraints that it satisfies and Chapter 10 compares it with other models extant in the literature. The dissertations ends with conclusions and future work in Chapter 11.

Chapter 2

Overview of ACT-R

The sentence-processing theory proposed in this dissertation was implemented in ACT-R (Anderson & Lebiere, 1998), which is a general theory of human cognition. In this chapter we describe those ACT-R concepts and mechanisms that are important for the understanding of the sentence-processing model.

ACT-R assumes that human knowledge is structured in two categories: declarative and procedural. The declarative knowledge refers to facts such as *Stockholm is the capital of Sweden* or $2 + 2 = 4$; these facts would be represented as **chunks** in ACT-R. The procedural knowledge corresponds to knowledge about carrying out actions (for instance about performing addition or about driving) and is expressed in ACT-R in the form of **productions**. ACT-R claims that human cognition is the effect of the interaction between procedural and declarative memory[1].

Chunks. ACT-R chunks encode "small, independent patterns of information" (Anderson & Lebiere, 1998); similar patterns of information are represented as chunks of the same **type**. For instance, facts such as *Stockholm is the capital of Sweden* and *Oslo is the capital of Norway* can be encoded in chunks of the same type. Chunk types are defined by ACT-R users; the definition specifies what kind of information all chunks of that type must carry in. The information within a chunk is structured in **slots**; the first slot (always called *isa*) records the type of the chunk. Table 2.1 shows the definition of the chunk type *capital*, having two slots — *country* and *city*; It also depicts two chunks of type *capital*: `Sweden-fact` and `Norway-fact`. For the `Sweden-fact`, `Sweden` is the **filler** of the slot *country* and `Stockholm` is the filler of the slot *city*. Chunks are characterized by their **activation**, which is a quantity reflecting how often and how recently the chunk was used in the past and how relevant it is to the current context. We talk later in this chapter about how chunk activation is computed.

[1] I use the terms *knowledge* and *memory* as interchangeable.

(chunk-type capital country city)

`Sweden-fact>`	`Norway-fact>`
isa capital	isa capital
country `Sweden`	country `Norway`
city `Stockholm`	city `Oslo`

Table 2.1: Sample ACT-R chunks.

Productions. A production is an if–then rule with a **condition** side, containing one or more conditions, and an **action** side, specifying a number of actions. If the conditions in the condition side are fulfilled, the production can be **fired** and the actions on the action side can be executed. Table 2.2 contains an example of production. The first condition of any production always refers to the current **goal**; the other conditions are typically **retrievals**.

The ACT-R concept of goal denotes a memory buffer holding one chunk that corresponds to the current focus of attention of the system[2]; ACT-R attempts to "satisfy" the current goal as soon as possible. A goal is satisfied when a production empties the goal buffer (this operation is called **goal popping**). ACT-R fires only productions whose goal conditions match the current goal. Thus, in the example in Table 2.2, the production *Answer-Capital-Question* could be selected to fire only if the current goal were a chunk of type *say-capital* with the slot *country* filled by some arbitrary value (denoted by the variable *=ctry*) and with the slot *answer* filled by *nil*.

The other condition type that can be specified in the if side of a productions is retrieval; it indicates that a chunk must be retrieved from memory and that it must match the pattern specified in the production. For instance, the production in Table 2.2 specifies that the chunk retrieved must be of type *capital* and that the filler of the slot *country* must be the chunk *=ctry*, the filler of the slot *country* in the goal. Thus, if the variable *=ctry* was bound to `Sweden`, then `Sweden-fact` would be acceptable for retrieval, but `Norway-fact` would not be.

If several productions match the current goal, ACT-R chooses one of them according to their relative **utility**. (This mechanism is called **conflict resolution**.) The production utility is calculated by the **Utility Equation**, which weighs the benefits of achieving the goal by firing the production versus the costs associated with firing it:

$$U_p = PG - C \tag{2.1}$$

where U_p is the utility of a production p; P is the expected probability of achieving the

[2] I use the terms "goal" and "focus (of attention)" as synonyms.

```
production Answer-Capital-Question
 if  =goal>
      isa say-capital
      country =ctry
      answer nil
     =country-fact>
      isa capital
      country =ctry
      capital =city
 then
     =goal>
      answer =city
```

Table 2.2: An ACT-R production. All names starting with = represent variables. The notation $=x>$ specifies that the following lines expand the slots of chunk $=x$.

goal, if the production p is fired; G stands for the goal value (i.e., for how much time should be spent for achieving that goal); and C is the expected cost (in time units) of achieving the goal, if production p is chosen.

If several productions match the goal, ACT-R chooses the production with the highest utility. If two productions have the same utility, one of them is chosen randomly[3],

Thus, cognition in ACT-R emerges from a set of productions that fire in a specified order; each production can retrieve information from memory and use it to modify the current goal such as to allow the production that pops it to be selected.

Chunk Activation. Before discussing how the activation of a chunk is computed, we need to introduce the concept of association between chunks. In ACT-R, chunks used together many times become associated. The strength of association S_{ij} between two chunks i and j measures how often chunk j was needed when chunk i was in the goal. Later, in Section 3.1.3, we see that the co-occurrence–based definition is not used in the sentence-processing model described in this dissertation, but it is replaced with a definition based on semantic similarity between words and propositions[4].

The activation of a chunk is decomposable into **base-level activation**, reflecting the

[3]Utilities are noisy quantities: to each utility, a noise with logistic distribution is added. The standard deviation of this noise can be specified by the modeler; in the simulations described in this dissertation it is always set to 0.09.

[4]The co-occurrence-based definition of associations has been recognized as one of the problematic aspects of ACT-R; the newest version of ACT-R does not make use of it anymore.

history of usage of a chunk, and **spreading activation**, reflecting its relevance to the goal. The more often or the more recently a chunk is used, the higher its base-level activation. The spreading activation depends on the other chunks that fill slots in the current goal; the more these chunks are associated with the chunk to be retrieved, the larger the amount of spreading activation. The ACT-R **Activation Equation** specifies how the activation of a chunk is computed:

$$A_j = B_j + \sum_{\text{goal slots } i} W_i S_{ij} \tag{2.2}$$

A_j is the total activation of chunk j; B_j is its base-level activation. The sum in the right-hand side of the equation is the spreading-activation component: each filler i of the goal slots spreads an amount of activation to the chunk j proportional with the association S_{ij} between chunks i and j. W_i is a weight reflecting how "attention" is split among all the elements currently in the goal: the higher the W_i of one goal slot, the more activation it spreads to chunks associated to it.

In ACT-R a chunk can be retrieved only if its activation is greater than a **retrieval threshold** level, τ. To model the variability of human cognition, ACT-R activations are noisy: a noise value is added to the magnitude computed by the Activation Equation 2.2. The noise comes from a logistic distribution with variance σ[5]. When there is no noise, if several chunks can be retrieved at a certain moment, the one with the highest activation (over the retrieval threshold) is chosen. However, if the activations are noisy, the probability to retrieve a chunk j is given by the **Retrieval-Probability Equation**:

$$P_j = \frac{1}{1 + e^{-(A_j - \tau)/s}} \tag{2.3}$$

with s being a quantity proportional to the noise variance σ: $s = \sqrt{3}\sigma/\pi$, τ the retrieval threshold, and A_j the activation of chunk j.

Latency. In ACT-R firing a production takes time; given that a cognitive task is fulfilled through a succession of production firings, the time for that task corresponds to the sum of the production latencies. The production latency can be split into two components:

1. The **effort** latency, covering the execution of the actions in the action side of the production
2. The **matching** latency, referring to the time needed for chunk retrievals[6]

[5]An ACT-R model has the choice of deciding whether activations are noisy or not by setting an activation noise parameter; that parameter, if not nil, gives the variance of the noise distribution.

[6]Not only does the matching component cover the retrievals made by the production fired, but also the unsuccessful retrievals made by productions tried before the current one.

From this way of dividing the production latency, we can see that there is a lower bound on its value: a production cannot take less than its effort component. The default value of the production effort is 50 ms; a production that performs time-consuming actions such as key presses can take longer, but typically productions take at least 50 ms. Hence, the more productions that fire to perform a task, the longer the overall time to complete that task.

Whereas the production effort is at least 50 ms, the matching latency can be infinitely small, depending on the activation of the chunks retrieved: the more active the chunks, the faster the retrievals. The ACT-R **Latency Equation** relates the time T_j to retrieve a chunk j to the activation A_j of that chunk:

$$T_j = Fe^{-(A_j + S_p)} \tag{2.4}$$

where F is a latency factor and S_p is the **strength of the production** that performs the retrieval (its default value is 0). The strength of a production S_p is a parameter quantifying how often the production was used; a production that was frequently fired takes less to retrieve a chunk than a production that was fired less often.

The Latency Equation has the potential for infinite latencies if the activation of a chunk is negative (and activations can be negative in ACT-R). However, because only chunks with activations above the retrieval threshold are retrieved, the retrieval latency is upper-bounded by the retrieval-failure latency, obtained by replacing A_j with the retrieval threshold τ in Equation 2.4. Thus, when the activation A_j of chunk j is less than τ, the retrieval of j fails and the retrieval latency is $Fe^{-f(\tau+S_p)}$. Consequently, retrieval failures take longer than any retrieval. If a production fails to retrieve a chunk, then that production cannot fire and the failure time is added to the latency of the next production fired.

Partial Matching. We saw that, for a chunk to be retrieved, it must match the conditions on the left-hand side of a production. Thus, in Table 2.2, only those chunks of type *capital* that encode the capital of the country *=ctry* in the goal can be retrieved. However, there are situations in which the exact-matching requirement is too strong — for instance, people can recognize mispronounced words or human faces that change over time. ACT-R can also be more resilient with respect to matching: if the partial-matching behavior is enabled, instead of trying to retrieve the chunk that matches exactly the retrieval condition, ACT-R will look for a chunk that resembles most closely, albeit not exactly, the retrieval condition. Thus, coming back to the example in Table 2.2, if *=ctry* was bound to Finland and, if there were no chunk about the capital of `Finland`, then it would be possible that the `Sweden-fact` be considered instead, provided that `Sweden` and `Finland` were quite similar. This choice would lead to ACT-R answering that `Stockholm` is the capital of `Finland`. However, to ensure that perfect matches are preferred to partial matches (i.e., that ACT-R does not respond `Stockholm` even if it knows very well that `Helsinki` is the capital of `Finland`), if the match is not perfect, the activation of the chunk retrieved will be discounted to reflect the degree of mismatch. Thus, if s is the similarity between `Finland` and `Sweden`, the discount will be $D * (1 - s)$, where D is a penalty constant. More generally, if a retrieval condition of production p specifies a value v_k for the slots k, then the **match score** of a chunk j with respect to production p is given by the following equation:

$$M_{pj} = A_j - \sum_{\text{conditions } k} D(1 - s_k) \tag{2.5}$$

s_k is the similarity between value v_k in the retrieval condition and the actual filler of slot k in chunk j; A_j is the activation of the chunk j. If slot k of j is v_k then the similarity is 1 and the corresponding discount is 0. The summation in Equation 2.5 involves only those slots k on which the production p actually imposes conditions.

When partial matching is used, the quantity M_{pj} replaces the activation A_j in the Latency Equation 2.4. Therefore, retrieving a chunk that is not a perfect match takes longer than the perfect match, provided that their initial activations A_i are the same. Also, if partial matching is enabled, as there are several chunks that compete for being retrieved, the probability of retrieving one of them depends on its activation and also on the activation of the other retrieval candidates:

$$P_{pj} = \frac{e^{M_{jp}/s\sqrt{2}}}{\sum_{\text{chunks } i} e^{M_{ip}/s\sqrt{2}}} \tag{2.6}$$

where P_{pj} is the probability of chunk j being retrieved by production p and s is, as in Equation 2.3, dependent on the noise variance σ: $s = \sigma\sqrt{3}/\pi$; the summation is done over all chunks of the same type (specified in the retrieval condition). Equation 2.6 is a generalization of Equation 2.3: when partial matching is turned off, the retrieval threshold is treated as the only chunk in competition with the chunk to be retrieved.

Base-Level Learning. We saw that chunk activation depends on base-level activation, which reflects frequency and recency of usage. Although the modelers can specify base-level values for the chunks in their ACT-R models, ACT-R itself has a mechanism for updating the base-level activations; this mechanism is called **base-level learning.** The **Base-Level–Learning Equation** defines the base-level activation B_i of a chunk i as a function of its history of usage:

$$B_i = ln(\sum_{j=1}^{n} t_j^{-d}) + \beta \tag{2.7}$$

$$\approx log\frac{n}{1-d} - dlogL + \beta \tag{2.8}$$

where t_j is the the time since the j-th usage of the chunk i, n is the number of times that the chunk was used, d is a decay factor, and β is a base-level constant. Sometimes, instead of the sum of powers, an approximation is used (Equation 2.8); in that approximation, L is the interval between the creation of chunk i and the current moment. The Base-Level–Learning Equation 2.7 says that (1) the more a chunk is practiced, the higher its activation (power law of practice); and (2) the longer the time since its last practice, the smaller its activation (power law of forgetting).

Production-Strength Learning. We saw that the Latency Equation 2.4 claims that the time for retrieving a chunk depends on the strength S_p of the production performing the retrieval. If that production is fired many times, the S_p parameter increases and, thus, the time spent for the retrieval decreases. The mechanism for updating production strengths is called **production-strength learning** and is captured by an equation similar to the Base-Level–Learning Equation 2.7:

$$S_p = ln(\sum_{j=1}^{n} t_j^{-d}) + \gamma \quad (2.9)$$

$$\approx log\frac{n}{1-d} - dlogL + \gamma \quad (2.10)$$

where t_j is the the time since the j-th firing of the production p, n is the number of times that the production was fired, d is a decay factor, and γ is a constant.

Summary. In conclusion, ACT-R is a cognitive architecture based on the distinction between declarative memory (chunks) and procedural memory (productions). At each moment, ACT-R attempts to satisfy the current goal, so it selects from the procedural memory one production that matches it and fires that production. The chosen production can retrieve declarative information from the memory and can update the goal or clear the goal buffer, thus finishing the task at hand. This symbolic level is supported by subsymbolic mechanisms that specify how ACT-R chooses which chunk to retrieve, how much time it takes for retrieval, or which production (among several matching the current goal) to fire.

Chapter 3

Comprehension of Isolated Sentences

This chapter describes how the sentence-processing model comprehends isolated sentences; later, in Chapter 5 we see how this basic model can be extended to account for verification and comprehension of sentences in discourse context.

As mentioned in Chapter 1, for each word, the sentence-processing model searches for a matching proposition in the long-term memory and then uses it as a candidate interpretation for the whole input sentence. In this process the model builds a propositional representation for the input and relates that representation to the sentence interpretation. Section 3.1 presents the meaning and propositional representations chosen for this model; Section 3.2 describes the comprehension algorithm.

3.1 Representation

The sentence processing model operates with **meanings** and **propositions**: it uses an **atomic meaning representation** and a **distributed propositional representation**. Chapter 8 contains a detailed discussion of the reasons behind these choices and of alternative representations.

3.1.1 Meaning Representation

The model represents the meaning of a word as an ACT-R chunk with no special slots. The **word-link** chunk (sometimes referred to as "word") groups together the lexeme[1] and the word meaning. Table 3.1 shows how the model represents the meaning and the lexeme of the word *child*. The chunk `*Child*` is a simple, unstructured ACT-R chunk and

[1]By lexeme we mean the string of letters forming a word.

`*child*>`	`Child-wlink>`
isa chunk	isa word
	lexeme `"child"`
	meaning `*child*`
	context `Experiment`

Table 3.1: Meaning and word-link representation for the word *child*.

stands for the meaning of *child*. The word link `Child-wlink` relates the meaning of *child* (slot *meaning*) and the string of letters that spell the word (slot *lexeme*); it also contains additional information about the context in which the word was learned (slot *context*).

When the model reads a lexeme, first it attempts to retrieve a word link that relates that lexeme to a meaning and, then, if successful, it examines the meaning in that word link. Thus, with such a representation, meaning extraction is a very simple process, which can be summarized in one ACT-R production (see Table 3.2). The production `extract-meaning` fires only in the meaning extraction phase (indicated by the value `''extract-meaning''` of the slot *task*). Initially, the slot *word* of the goal contains the lexeme of the current word; after meaning extraction, the lexeme is replaced by the actual meaning (see the right-hand side of the production *extract-meaning*). The right-hand side of the production also creates a new link chunk connecting the current word meaning (`=mn`) to the proposition `=goal` corresponding to the current input sentence.

3.1.2 Propositional Representation

Unlike for meanings, my model uses a **distributed representation for propositions** — that is, it represents separately the concepts that occur within a proposition, rather than grouping them together in a single chunk. This kind of representation is consistent with a new trend towards fragmented, hierarchical representations in the ACT-R community, exemplified by studies such as Anderson, Bothell, Lebiere, and Matessa (1998), Salvucci and Anderson (2001), Anderson, Budiu, and Reder (in press). Figure 3.1 shows how the model represents the proposition *Noah took the animals on the ark*: the proposition is encoded in the node *Ark Prop* and the labeled links connect the proposition chunk to the concepts involved in that proposition. Thus, the *agent* link relates the proposition node *Ark Prop* to the agent of the proposition, **Noah**; the *verb* link relates *Ark Prop* to the concept that is the verb of that proposition, **take** and so on. Figure 3.1 also shows the chunk corresponding to one of the links. The links contain all the information pertaining to the structure of the proposition: they keep track of the nodes that they connect (slots *parent* and *child*), of the type of the link (slot *type*) and of the context in which they were last used (slot *context*). In addition, the slot *interpretation* points to the long-term memory

```
production extract-meaning
 if  =goal>
       isa comprehend
       task ``extract-meaning''
       word =lex
       role =role
     =word-link>
       isa word
       lexeme =lex
       meaning =mn
 then
     =goal>
       word =mn
       task ``interpretation''
     =sent-link>
       isa prop-link
       type =role
       parent =goal
       child =mn
       context experiment
```

Table 3.2: The ACT-R production used for extracting the meaning of a word.

interpretation (if any) that was found for the input sentence at the time of processing the corresponding concept. Note that in the propositional representation concepts, rather than lexemes, appear.

For each input sentence, the model produces a propositional representation[2]. The representation contains a pointer to a long-term–memory proposition that matches the input sentence (the final sentence interpretation). Thus, comprehension involves (1) building a propositional representation for the input sentence, and (2) relating that representation to another proposition in the long-term memory (or in the discourse).

3.1.3 Semantic Overlap

As discussed in Chapter 2, ACT-R knowledge can be either procedural (productions) or declarative (chunks). An ACT-R model makes use of both forms of representation. In

[2]The syntactic and verbatim representations are not considered by this theory.

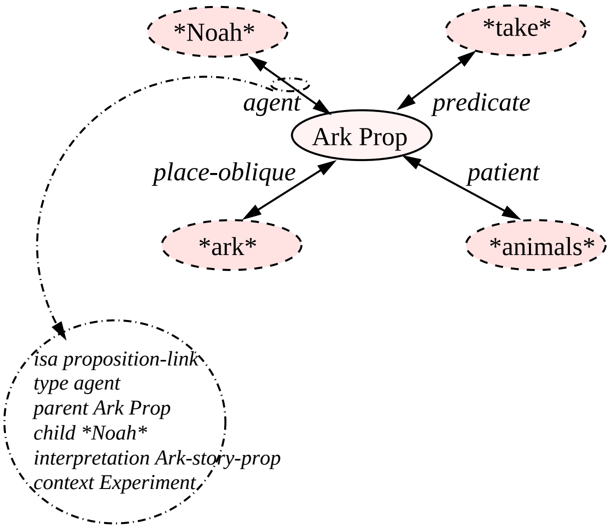


Figure 3.1: Representation of the proposition *Noah took the animals on the ark.*

Chapter 1 we saw that three aspects characterize the sentence-processing model: search for an interpretation, matching of the candidate interpretation against the input sentence, and interpretation priming. All of these aspects can be implemented relying mainly on procedural knowledge (see also the discussion in Chapter 8 on modeling alternatives) — that is, the model could pick up one arbitrary proposition from background knowledge, then look at its components and check explicitly whether they match the input; if they did not, the model could try another proposition until it found a matching one. Such a heavily-procedural process would use no heuristics and would need to explore fully the search space to find the correct interpretation, thus taking an amount of time polynomial in the number of propositions in the search space. However, in Section 1.4.2, we saw that a sentence-processing model must be fast and scalable, so it should not depend on the number of propositions in the long term memory.

One way to inform the search process is by spreading activation from the current words in the focus to those propositions that are relevant. In Chapter 2 we discussed the ACT-R mechanism of spreading activation from the chunks in the goal to other related chunks and we saw that the amount of activation spread depends on the associative strength between the two chunks.

Traditionally, in ACT-R the associations between chunks are positive or negative magnitudes that reflect co-occurrence in the same context (Anderson & Lebiere, 1998). Unlike ACT-R, my model posits that associations express semantic similarity between concepts and/or propositions. The model takes similarities between word meanings as preset values[3]

[3] An exception occurs when the model learns new meanings; in that situation it also learns similarities

between 0 and 1. Although the meaning similarity settings are qualitatively reasonable, the exact values of the similarities used by the model are based on intuition, rather than on similarity ratings. Therefore, I will strive to show that the ordinal predictions of the simulations depend only on the assumed orderings of these similarities, which are quite defensible.

Next I present how similarities between propositions, between propositions and meanings and between proposition links and meanings are computed, starting from meaning similarities. In all my definitions, similarity is symmetric (i.e., if the similarity between A and B is σ, then the similarity between B and A is also σ).

Similarity between a meaning and a proposition. Suppose the concepts $c_1 \ldots c_n$ occur in the proposition p. Then the similarity $\sigma(m, p)$ between a meaning m and the proposition p made of those concepts is defined as follows:

$$\sigma(m, p) = \frac{1}{n} \sum_{i=1}^{n} \sigma(m, c_i)$$

where $\sigma(m, c_i)$ is the similarity between the meaning m and the concept c_i. This definition says that the similarity of a meaning to a proposition is a weighted sum of the similarities between that meaning and each of the concepts involved in the proposition.

Similarity between a meaning and a proposition link. Remember that a propositional link is a labeled link in the graph in Figure 3.1. It connects a proposition to one concept involved in that proposition, and it is labeled with the thematic role of that concept. We have seen that a propositional link is an ACT-R chunk: the slot *parent* contains the interpretation and the slot *child* contains a concept in that interpretation. I define the similarity between a link and a meaning as being the similarity between the meaning and the concept in the slot *child* of the link.

Similarity between two propositions. Assume that proposition p_1 is made of concepts $c_{11} \ldots c_{1n}$ and proposition p_2 is made of concepts $c_{21} \ldots c_{2n}$ and, moreover, that c_{1i} and c_{2i} have the same thematic role (i.e., agent, patient, etc.), for all i. (If one proposition has thematic roles not shared by the other, they can be ignored in computing the similarity). Then the similarity $\sigma(p_1, p_2)$ between p_1 and p_2 is defined as follows:

$$\sigma(p_1, p_2) = \frac{1}{n} \sum_{i=1}^{n} \sigma(c_{1i}, c_{2i})$$

between the new meanings and the other meanings.

where $\sigma(c_{1i}, c_{2i})$ is the similarity between concepts c_{1i} and c_{2i}. Thus, the similarity between two proposition is a weighted sum of the similarities between fillers of corresponding thematic roles in the two propositions. Note that it is important to look at similarities between corresponding roles: if the role information were not taken into account, then *Bill hit Tom* and *Tom hit Bill* would have similarity 1, because they share all concepts, although they distribute them differently to thematic roles. With our formula, assuming there is no similarity between the concepts corresponding to *Tom* and *Bill*, the similarity between the two propositions is only $\frac{1}{3}$. A less extreme (and perhaps more realistic) view would be that similar words in different thematic roles do bring a contribution to the overall similarity between propositions, but that contribution is smaller than if they had the same thematic role.

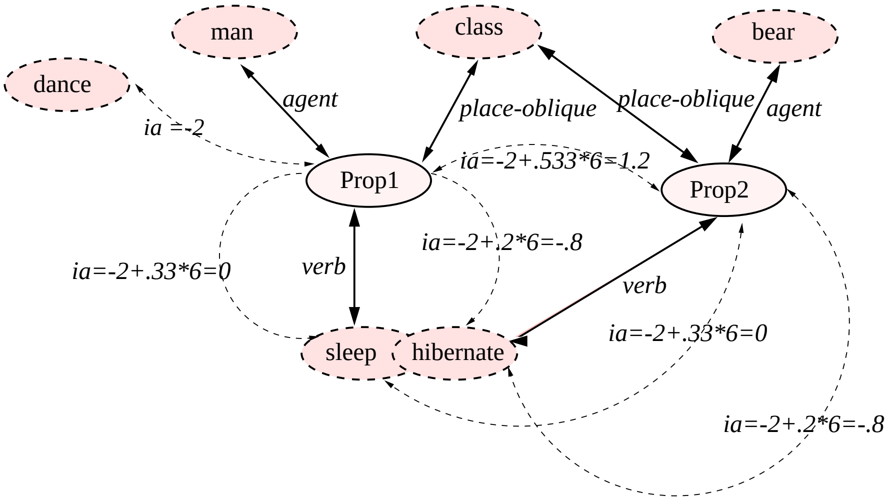
Similarity between a proposition and a proposition link. As for meanings, the similarity between a proposition and a proposition link is defined as the similarity between that proposition and the meaning filling the slot *child* of the link.

Given the similarity between two chunks i and j, I calculate the associative strength S_{ij} between them using a linear function of the similarity:

$$S_{ij} = B_{ia} + i_{ia} * \sigma(i, j) \tag{3.1}$$

where B_{ia} is a base associative strength and i_{ia} is a positive increment. B_{ia} is assumed to be a negative quantity, indicating that two concepts and/or propositions can be positively associated only if they are similar enough. Note that, although traditionally associative strengths are not symmetric in ACT-R (i.e., $S_{ij} \neq S_{ji}$), they become symmetric if Equation 3.1 is used to set them, because similarity is symmetric.

Figure 3.2 shows an example of how associative strengths are computed. First, let us look at the semantic overlap between the proposition *The man sleeps in class* and two different concepts, *sleep* and *hibernate*. The similarity between *sleep* and the proposition *The man sleeps in class* is $\frac{1}{3}(0 + 1 + 0) = 0.33$, assuming that *sleep* bears similarity 0 to *man* and *class*. Then, for a setting of $B_{ia} = -2$ and $i_{ia} = 6$, according to Equation 3.1, the associative strength between *sleep* and the proposition *The man sleeps in class* is $-2 + 0.33 \times 6 = 0$. We can also compute the similarity between *The man sleeps in class* and the concept *hibernate*, if we know the similarity between *hibernate* and all the other concepts in the proposition. Suppose the similarity between *sleep* and *hibernate* is 0.6 and the similarity between *hibernate* and the other concepts in the proposition (i.e., *man, class*) is 0. Then the similarity between *The man sleeps in class* and *hibernate* is $\frac{1}{3}(0 + 0.6 + 0) = 0.2$, and the associative strength between *hibernate* and *The man sleeps in class* is $-2 + 0.2 \times 6 = 0.8$. Note that the similarities between *hibernate* and the verb link of the proposition *The man sleeps in class* is the same as the similarity between *hibernate* and *sleep*, i.e 0.6.



Similarity (sleep, hibernate)= 0.6 (preset)
Similarity (sleep,Prop1) = 1/3 = 0.33 — *Similarity (sleep,Prop2) = 0.6/3 = 0.2*
Similarity (hibernate, Prop1) = 0.6/3=0.2 — *Similarity (hibernate, Prop2) = 1/3=0.33*

Similarity (Prop1, Prop2) = 1/3(0+0.6+1)=0.533*

Figure 3.2: Representation of the proposition *The man sleeps in class*. The associative strength between the proposition and various concepts is given by the formula $B_{ia} + \sigma * i_{ia}$ with $B_{ia} = -3$ and $i_{ia} = 1$.

If we want to compute the similarity between two propositions, *The man sleeps in class* and *The bear hibernates in class*, we can use the similarities between corresponding thematic roles. Thus, if there is no similarity between *man* and *bear* and if there is 0.6 similarity between *sleep* and *hibernate* and if the similarity between the two usages of *class* in the two propositions is 1, then the similarity between *The man sleeps in class* and *The bear hibernates in class* is given by $\frac{1}{3}(0 + 0.6 + 1) = 0.533$. Therefore, the associative strength between the two propositions is $-2 + 0.533 \times 6 = 1.198$, if B_{ia} and i_{ia} are set as before. Also, the similarity between *The man sleeps in class* and the *verb* link of the proposition *The bear hibernates in class* is the same as the similarity between the first proposition and *hibernate*, namely, 0.2.

The effect of having associative strengths between concepts and propositions is that, according to activation equation 2.2, if one concept, say *sleep*, is in the focus of attention, then it will spread activation to all the propositions in memory. The activation spread is proportional to the associative strength S_{ij} between the concept in the focus and the proposition. Thus, according to equation 3.1, because $B_{ia} < 0$, most of the activation spread is negative, unless the proposition is sufficiently similar to *sleep*. In that latter case, the proposition actually benefits from the presence of *sleep* in the focus and is more likely to be retrieved. Thus, these associations and the spreading activation process serve to focus processing on a relatively small part of the data base.

This section started with a discussion about the task of choosing the right amount of procedural knowledge. Such a choice depends on the data to be modeled; for instance, given the ACT-R assumption that each production cannot take less that 50 ms for firing, the length of the execution paths in the model gives a lower bound for the latency of the total computation. The motivation for reducing the search was exactly that comprehension happens too fast for a too complex search to take place. The size of the associations between chunks affect the latencies through retrieval times, but those retrieval times can be made arbitrarily small by manipulating a number of ACT-R parameters (e.g., F in the latency equation 2.4 or B_j in the activation equation 2.2 in Chapter 2). Thus, whereas it is possible to have a relatively small latency for an informed search process in which few productions fire and the correct interpretation is found with the help of associations, it is harder to satisfactorily limit the number of productions firing in an uninformed search.

3.2 Model Description

This section shows how the model comprehends isolated sentences; in Chapter 5 we see how to modify this model to account for comprehension of sentences within a text.

The aim of the sentence-processing model is to "guess" the meaning of the current sentence as soon as possible. This aim derives from the assumption of incrementality of language comprehension, which is supported by a number of experimental studies (Marslen-Wilson, 1973, 1975; Tyler & Marslen-Wilson, 1982; Oakhill, Garnham, & Vonk, 1989; Traxler, Bybee, & Pickering, 1997). These studies show that basic aspects of sentence comprehension are performed on line. The most famous are perhaps Marslen-Wilson's studies with close shadowers, who can repeat speech they hear over headphones at a delay of one syllable. Close shadowers correct errors that they hear (for instance, pronunciation errors), but only if the words are part of a syntactically and semantically well-formed sentence. On the other hand, there is evidence that people take longer to read the sentence (or clause) endings than it would be expected on the basis of their lexical content (Mitchell & Green, 1978; Green, Mitchell, & Hammond, 1981).

My model is highly incremental: with each new content word, it attempts to find a sentence interpretation consistent with all the words seen up to that moment. However, in agreement with the findings of Mitchell and Green, at the end of the sentence, the model performs an integration phase, in which it attempts to relate the representation for the input sentence to the structures extant in memory.

A summary description of the model contains the following basic steps:

Start with no candidate interpretation.

1. **Read.** Read next word.

2. **Search.** If there is no candidate interpretation, search for one; if none is found go to the next word (step 1).

3. **Match.** If the candidate interpretation matches the current word, go to the next word (step 1); else go to either step 1 or step 2.

   Integration. At the end of the sentence, integrate the current sentence with its interpretation, if any.

Chapter 1 identified three aspects of the sentence processing model: the search for an interpretation, the match between the input and the propositions in the long term memory, and the participation of candidate interpretations in the comprehension process. Search and match have each of them one individual step assigned in the preceding description of the model. Candidate interpretations proven invalid may help at finding new candidates in step 2. At the end of the sentence, in the integration phase the model must make sure that the structures created are consistent. Let us take a closer look at each of the these aspects of the model.

Search. The search process is procedurally simple: it selects the long-term–memory proposition that has the highest activation above the retrieval threshold. If there is no such proposition, the model goes on to the next word. Otherwise, depending on the outcome of the matching process (step 3), the candidate interpretation is either accepted or rejected. In the case of rejection, the model marks that proposition as "visited" and either continues the search by looking at the next best proposition or goes on to the next word. The decision to stop the search is probabilistic and the probability may vary from task to task or from individual to individual. If the model decides to move to the next word without having found an interpretation, it creates a chunk called **bug**, which registers the failure to find an interpretation and some extra information about the context in which the failure occurred (e.g., current word, current thematic role, previous candidate interpretation). Bugs are used in verification tasks to decide whether the sentence may be false or in comprehension of sentences conveying novel information or containing new words (see Chapter 5).

In Section 3.1.3 we discussed the need for an informed search process, namely for a search process that uses most of the information available to speed up the finding of the right interpretation. Because people are very fast to comprehend sentences, the model should find the right candidate interpretation as soon as possible in the search process: each failure costs time[4]. One necessary condition for speeding up the search process is that the

[4] Each time the model finds a wrong interpretation, it must spend extra time to look for another candidate interpretation.

```
production find-interpretation
 if  =goal>
       isa comprehend
       word =word
       word-1 =word-1
       word-2 =word-2
       previous-interpretation =prev
       interpretation none
     =int>
       isa comprehend
       - context experiment
       - last-user[a] =goal
 then
     =goal>
       interpretation =int
     =int>
       last-user =goal
```

```
production stop-search
 if  =goal>
       isa comprehend
       word =wd
       role =role
       task ''interpretation"
       interpretation none
       previous-interpretation =prev-int
 then
     =goal>
       task "read"
       interpretation none
       previous-interpretation none
     =bug>
       isa bug
       word =wd
       role =role
       context =goal
       interpretation =prev-int
```

[a]In the actual model we use the *word* slot to keep the last user of an interpretation.

Table 3.3: Productions involved in the search for an interpretation.

activation of the right interpretation be higher than the activation of wrong propositions. To achieve this condition, the model uses spreading activation from the current focus. At each moment, the model keeps the last three word meanings processed in the focus; these meanings should occur in the correct interpretation of the sentence and, therefore, they should be highly similar to that interpretation. But, in this model, similarity entails associative strength; thus, the presence of these meanings in the focus raises the activation of the propositions that contain them (or meanings similar to them) and, hence, the activation of the interpretation sought by the model.

Table 3.3 presents the two productions involved in the actual search process. The production *find-interpretation* has a higher utility than the other production, *stop-search*, so it is tried first most of the time. It attempts to retrieve the most active interpretation not used before[5]. As mentioned previously, the last three words read, being in the focus (in

[5]The negative test on the slot *last-used* performs this function. Additionally, there is a negative test on

slots *word, word-1, word-2*), raise the activation of the propositions matching them, through the spreading-activation mechanism. Also, the previous candidate interpretation (proven invalid) is in the focus and spreads activation to related propositions; in the discussion on interpretation priming I explain why this behavior is desirable. When *find-interpretation* is successful, the interpretation retrieved is marked as used by setting its slot *last-user* to the value of the current goal.

If no interpretation is above the retrieval threshold, the production *find-interpretation* will fail and the other production, *stop-search* will fire. *Stop-Search* creates a bug recording the search failure and moves on to reading the next word[6].

Matching. We saw that a candidate interpretation is accepted only if it matches the current word. Table 3.4 shows the two productions responsible for matching: *successful-match* and *failed-match*; the former has a higher utility than the latter, so it is fired most often. In this model, matching is defined based on similarity: if the current word is similar enough to the corresponding concept in the interpretation, then the matching is successful. Production *successful-match* captures that definition: to check whether the current word `=word` with the thematic role `=role` (e.g., agent) matches the candidate interpretation, it attempts to retrieve the link labeled `=role` from the interpretation `=int`. Matching involves comparing `=word` with the filler `=child` of the slot *child* in the link `=int-link`, that is, comparing the current word with the concept having the same thematic role in the candidate interpretation. However, asking for `=word` and `=child` to be identical would be too strong a constraint; for instance, if the current word `=word` were *Tom* and the interpretation involved the concept *man* (i.e., if `=child` were bound to *man*) such a verification would fail. Instead, the model simply attempts to retrieve the link `=int-link`: success of retrieval is equated with matching, failure with nonmatching. The argument behind this definition of matching is that, if the similarity between the current word `=word` and the corresponding concept (`=child`) in the interpretation is high enough, then the activation spread from the current word, which is in focus, to `=int-link` will increase the overall activation of `=int-link` above the retrieval threshold. Otherwise, if the word and the concept `=child` in the interpretation are not similar enough, the activation of `=int-link` will remain under the retrieval threshold, and `=int-link` will not be retrieved. In this case, production *successful-match* fails and *failed-match* fires instead. The latter indicates the rejection of the current interpretation by assigning the value `none` to the slot *new-interpretation* of the goal[7].

---

the slot *context*, ensuring that the selected proposition is not some proposition read within the experiment. The latter test is only used for comprehension of isolated sentences.

[6] The production *Stop-Search* performs more bookkeeping and an extra word retrieval in the actual model; however, those details are not essential for understanding its basic behavior.

[7] When *successful-match* fires, it sets the slot *new-interpretation* to the accepted candidate interpretation.

```
production successful-match
 if  =goal>
       isa match
       role =role
       word =word
       interpretation =int
       new-interpretation nil
     =int-link>
       isa prop-link
       parent =int
       child =child
       type =role
       - context experiment
 then
     =goal>
       new-interpretation =int
   !pop!
```

```
production failed-match
 if  =goal>
       isa match
       role =role
       word =word
       interpretation =int
       new-interpretation nil
 then
     =goal>
       new-interpretation none
   !pop!
```

Table 3.4: Productions for matching an interpretation against the current word.

Interpretation priming. During the search process, beside the last three meanings processed, the model keeps in focus the candidate interpretation. If one such candidate does not match the current word meaning, then it is invalidated, but kept around in the focusgoal (in slot *previous-interpretation* in the goal — see Table 3.3) during the search for a new candidate interpretation. Thus, for a limited amount of time[8], through activation spreading, an invalid candidate can favor the choice of other propositions to which it is similar. Indeed, a candidate that was valid for many words, but is suddenly proved inadequate is like a synthetic memory of the words in the sentence that are no more in the focus. Those words have no other way of influencing the choice of an interpretation, but through such a candidate that matched them. Presumably, the right interpretation is among other propositions similar to the rejected candidate.

However, this game is dangerous: it can prevent the model from finding the right interpretation, if it is not close enough to the rejected candidate (e.g., because it differs in concepts that have not been processed yet). Indeed, if there is a cluster of wrong propositions near the rejected one, those may be preferred to the right proposition, which

[8]If no interpretation is found, the previous candidate interpretation gets out of the goal and cannot influence further choices.

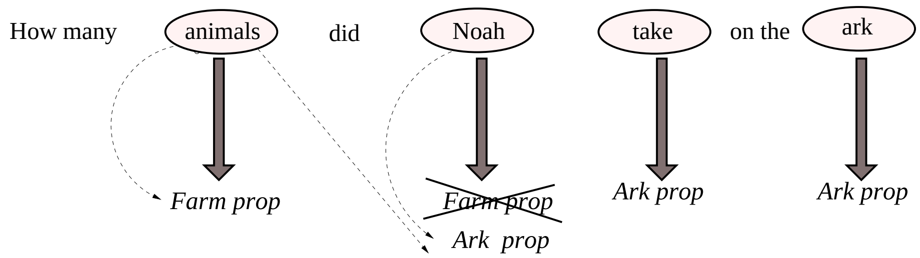


Figure 3.3: Comprehension of the sentence *How many animals did Noah take on the ark.*

may be never discovered[9]. Therefore, the influence of past candidates must be limited: if no valid candidate is found on the next try, then the past candidate simply gets out of focus. Specifically, in production *stop-search* in Table 3.3, on failing to find any interpretation that matches the current word, the model resets the value of the goal slot *previous-interpretation*.

In a related vein, given that an old interpretation is a history of previous words, we can ask the question whether the model could work using only the current word and an old interpretation. This is a solution unlikely to be successful: beside the previous-word information, the old interpretation also contains incorrect concepts (corresponding to the word that lead to its rejection and to all other concepts that did not appear in the input) and the model has no way of distinguishing between the relevant and the irrelevant information carried within the old interpretation. On the other hand, it is true that the current variant suffers from the same confusion, although in a more attenuated form (because the previous two words are in the focus and tip the balance towards the interpretations matching them rather than towards those matching the irrelevant information in the old interpretation).

Figure 3.3 shows how the model comprehends a simple sentence such as *How many animals did Noah take on the ark?* First, it processes the meaning of the patient *animals* and, with that concept in focus, starts looking for an interpretation. All propositions containing the same concept get an activation bonus from the word *animals*. One of these propositions is picked up; let it be *Father raises animals on the farm*. Next, the model must check whether this proposition actually matches the current word (*animals*); namely, whether the patient of the farm proposition is similar enough to the concept *animals*. In this case it is, so the model validates the farm proposition as a candidate interpretation and goes to the next word, the agent *Noah*. The model verifies whether *Noah* matches the agent of the current candidate interpretation; if *father* and *Noah* are dissimilar, then the farm proposition is invalidated and another candidate proposition must be found. This time, both *animals* and *Noah* (together with the farm proposition) are in the focus, so they spread activation towards propositions that are similar to them. The activation spread from

[9]Remember that the model never searches exhaustively: after an arbitrary number of failures, it can decide to stop processing of the current word.

```
production integrate
 if  =goal>
       isa comprehend
       interpretation =int
       - interpretation none
       task "integrate"
     =sent-link>
       isa prop-link
       parent =goal
       - interpretation =int
       context experiment
 then
     =sent-link>
       interpretation =int
```

```
production end-integration
 if  =goal>
       isa comprehend
       task "integrate"

 then
   !pop!
```

Table 3.5: Productions for integration.

the sources combines additively, such that the proposition that is most similar to all chunks in the focus gets retrieved. Let us assume that this proposition is *Noah took the animals on the ark*. Because *Noah* matches the agent of the ark proposition, this proposition becomes a new candidate. Next the model checks whether the other words *take* and *ark* match the ark proposition and, because they do, that is the final interpretation of the sentence.

Integration. With each new word that it processes, the model builds a new sentence link in the propositional representation of the input. Coming back to the example in Figure 3.3, at the end of the sentence, the representation for the input *How many animals did Noah take on the ark?* looks as in Figure 3.4(a). For each thematic role there is a separate sentence link, but, whereas all links created after finding the correct interpretation of the sentence (i.e., the ark proposition) contain a pointer to that proposition (in slot *interpretation*), the links built before considering it (i.e., the link corresponding to *animals*) bear inaccurate information in the slot *interpretation* (in our example, the *patient* link points to the farm proposition, candidate subsequently invalidated). To make the representation consistent, the model needs to retrieve those links that contain incorrect interpretation pointers and modify them. This process is termed **integration**; the productions that implement it are shown in Table 3.5. The production *integrate* retrieves the inconsistent links and updates them. The production *end-integration* has smaller probability of firing and stops the process of integration when there are no more incorrect links. Figure 3.4(b) shows the propositional representation of the input sentence after integration.

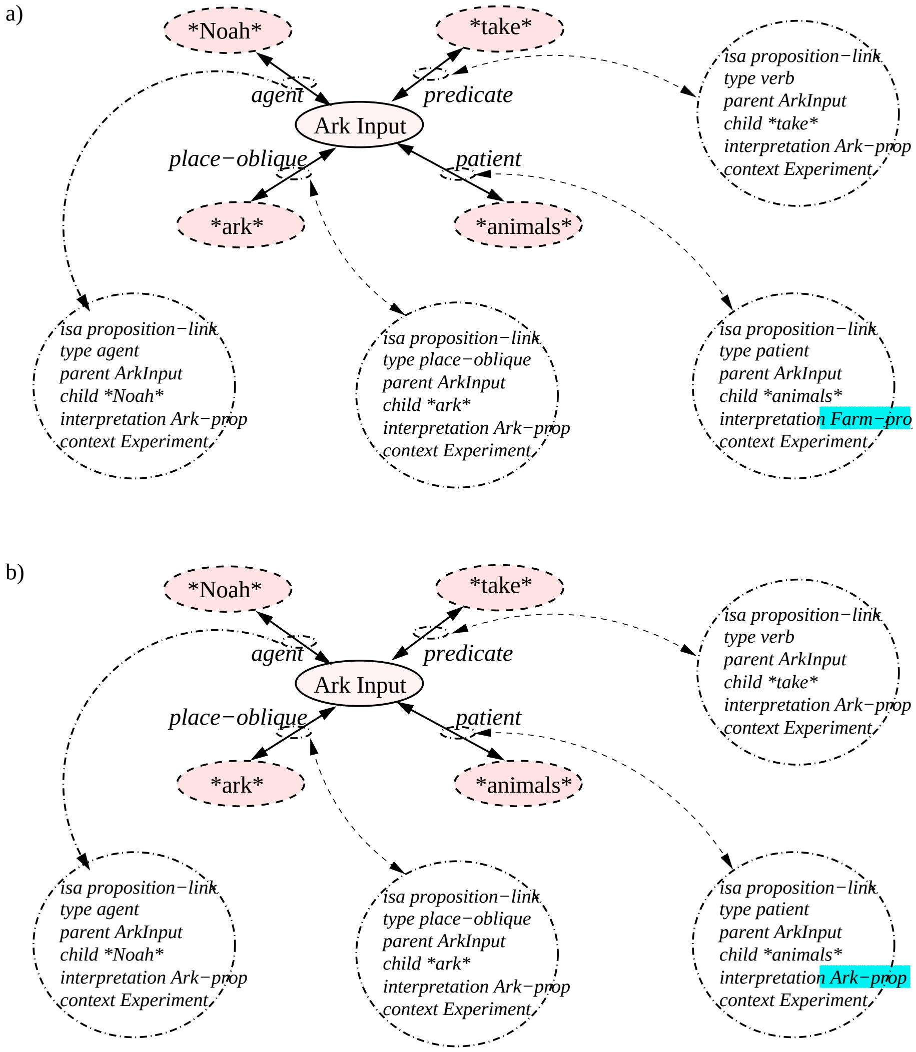


Figure 3.4: Propositional representation of the input sentence *How many animals did Noah take on the ark?*. a. Before integration. b. After integration.

Even though the integration phase in this model offers an account of the wrap-up processes that may occur at the end of sentence (Mitchell & Green, 1978; Green et al., 1981), this account is by no-means complete. More complex processes (e.g., monitoring goals, drawing causal inferences) may take place in the integration phase; however, they are not modeled explicitly. The basic assumption behind the integration phase is that the cognitive system spends some time at the end of the sentence (or clause) to relate the current sentential input to prior (episodic or permanent) knowledge. In Chapter 5 we see that the integration happens for both old sentences (for which there is an interpretation in the long-term memory) and for novel sentences (for which the model finds an approximative interpretation, rather than a perfectly matching one).

Chapter 4

Empirical Evaluation: Isolated Sentences

Chapter 3 showed how the sentence-processing model works on isolated sentences. Before discussing how this basic model can be extended to comprehend sentences embedded in a context (in Chapter 5), I use the isolated-sentence model to simulate two types of empirical phenomena: position effects on comprehension of isolated metaphoric sentences and Moses illusions.

4.1 Position Effects on Metaphor Comprehension

In the preceding chapters the term "metaphor" was used many times, without being ever defined. The Merriam-Webster's Dictionary defines it as "a figure of speech in which a word or phrase literally denoting one kind of object or idea is used in place of another to suggest a likeness or analogy between them (as in *drowning in money*))". Psycholinguists use it in a broader sense, covering also part of what Merriam-Webster calls "simile": "a figure of speech comparing two unlike things that is often introduced by *like* or *as* (as in *cheeks like roses*)" — a simile with no comparison term (such as *like* or *as*) is considered a metaphor. Thus, *A sonnet is a moment's monument* (D.G. Rossetti) or *Presentiment is that long shadow on the lawn* (E. Dickinson) are metaphors. The metaphor involves two terms: the **topic**, which is the object or idea spoken of, and the **vehicle**, to which the topic is compared. The topic in Rossetti's metaphor is *sonnet* and the vehicle is *(moment's) monument*. *A is B* metaphors such as Rossetti's are called **predicative metaphors**. Another type of metaphor is **anaphoric metaphor**, in which the topic and the vehicle do not appear in the same sentence, but the vehicle is used to refer to a topic previously introduced (or hinted at) in the discourse; *The hermit withdrawn to himself, avoiding the settlements, Sings by himself a song* (W. Whitman) is an anaphoric metaphor — *the hermit* refers to a bird.

Metaphor-first	Metaphor-last
Drops of molten silver filled the night sky	*The night sky was filled with drops of molten silver*
The parallel ribbons were followed by the train	*The train followed the parallel ribbons*
An angry cloud contorted his face	*His face was contorted by an angry cloud*
Stitches were left in the sand by the running birds	*The running birds left stitches in the sand*
Great woolly mushrooms surrounded the airplane	*The airplane was surrounded by great wooly mushrooms*

Table 4.1: Sample materials used by Gerrig and Healy (1983).

There is psycholinguistic evidence that people find the predicative metaphors to be easier to understand than anaphoric metaphors. However, even within the same class of metaphors, there are differences between the ease of comprehension of various metaphors. Some of the properties that facilitate metaphor comprehension are intrinsic to the metaphor (for example, goodness and familiarity); others, however, are properties of the context in which they appear. Thus, Ortony, Schallert, Reynolds, and Antos (1978) showed that metaphoric sentences preceded by a long and supportive context can be read as fast as literal sentences; however, this finding did not hold when the long context was replaced by a short, less informative one. Inhoff, Lima, and Carroll (1984) followed up Ortony et al.'s investigation and determined that the supportiveness of the context (rather than its length) was crucial for the ease of metaphor comprehension: even when metaphoric sentences were preceded by a short context, subjects read them as fast as literal sentences, if the context was supportive. Even at the sentence level, a context that precedes the metaphor can facilitate the comprehension compared to a context that follows the metaphor (Gerrig & Healy, 1983). This section discusses in detail this effect of sentence context on metaphor comprehension and describes how the sentence-processing model from Chapter 3 can simulate it.

4.1.1 Behavioral Data (Gerrig and Healy, 1983)

Gerrig and Healy (1983) showed that the position of the metaphor within a sentence may influence the speed of comprehension. They presented their subjects with two kinds of sentences: sentences starting with a metaphor and sentences ending with a metaphor; one type of sentence was usually obtained by making the other passive. Table 4.1 contains some examples of metaphor-first and metaphor-last sentences.

Gerrig and Healy measured reading times for metaphor-first and metaphor-last sentences and found that subjects read metaphor-first sentences slower than metaphor-last

	Reading Times	
Type of sentence	Data	Model
metaphor-first	4.21	4.30
metaphor-last	3.53	3.68

Table 4.2: Mean reading times (s) for metaphorical sentences from Gerrig & Healy (1983): data and model.

sentences. To make sure that this result was not an artifact of the different sentence structure of the two types of targets, Gerrig and Healy ran a second experiment in which they introduced two literal conditions, obtained by replacing metaphors with equivalent literals in the two types of target sentences. Whereas, for metaphors, the second experiment replicated the results from the first experiment, no distinction was found between the reading times in the two literal conditions, thus indicating that the difference between metaphor-first and metaphor-last sentences was not caused by the structure of targets. Table 4.2 presents the reading times of the metaphoric targets in the two conditions from the first experiment in Gerrig and Healy (1983). This result is a nice demonstration that people dynamically interpret and reinterpret the sentence as they read it. If they waited until the end to assign an interpretation to the sentence, there should be no difference between the two conditions. Thus, the existence of a difference supports a key assumption of my processing model: incrementality (see Sections 1.1 and 3.2).

4.1.2 Simulation of the Metaphor-Position Effect

Let us take a look at how the sentence-processing model behaves on Gerrig and Healy's data. First, we consider metaphor-first sentences. Figure 4.1(a) shows the sequence of interpretations for the sentence *Drops of molten silver filled the night sky*: the first words (*Drops of molten silver filled*) suggest that the sentence may be about a container holding liquid silver, but the final words (*night sky*) do not match such an interpretation. Therefore, the model must reject the container interpretation and find a new candidate interpretation, which could be the correct interpretation *Stars fill the night sky*, provided that *stars* and *drops of molten silver* are similar enough. But switching to a new interpretation costs the model extra time. On the other hand, such a switch happens less often in the case of metaphor-last sentences. For such a sentence (Figure 4.1b) it is more probable that, after reading *The night sky was filled with*, the model selects the correct stars interpretation. The stars interpretation would be then validated by the last words of the sentence (*drops of molten silver*). Thus, the model predicts that metaphor-first sentences take longer than metaphor-last sentences, because, for the former, one candidate interpretation must be rejected and replaced with another one. The latency results produced by the model are

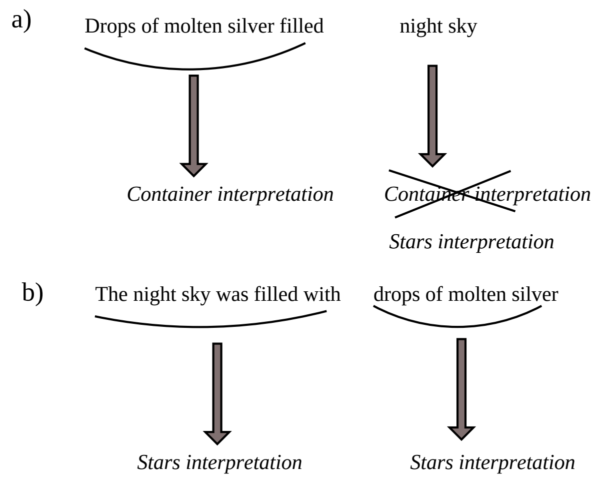


Figure 4.1: Processing of metaphoric sentences.

given in the third column of Table 4.2.

The model treats word phrases such as *drops of molten silver* as one meaning and does not act on each component of them. Moreover, the model does not attempt to build the meaning of a noun phrase — it assumes that there already is a meaning for *drops of molten silver*. Such an assertion is certainly not true, even for noun phrases used by Gerrig and Healy. For instance, the existence of an atomic meaning for *great woolly mushrooms* is very unlikely. One possible solution to this problem is to have the model run recursively on noun phrases: when it recognized the beginning of a noun phrase, it could abandon temporarily the comprehension of the whole sentence and attempt to find an interpretation for the noun phrase. A consequence of treating noun phrases as atomic meanings is that they get to be in focus as a whole. Thus, literally speaking, if only the last three content words were kept in focus, when the word *sky* was processed in the sentence *Drops of molten silver filled the sky*, the words in focus should be *sky, filled,* and *silver*. However, the goal slots are filled by *drops-of-molten-silver, filled,* and *sky*, because the model keeps in focus concepts rather than words.

Critical to the predictions of the model is the significantly smaller chance of an interpretation switch for metaphor-last sentences. The model's basis for capturing the latency pattern in Gerrig and Healy (1983) is that metaphor-first sentences are reinterpreted once more at the last concept (*the night sky*), whereas metaphor-last sentences do not need a reinterpretation in most cases. This difference is a consequence of the character of knowledge in the long term memory: if there are few propositions matching *The night sky* or *The night sky was filled*, then there will be a high chance that the right interpretation for the sentence *The night sky was filled with drops of molten silver* is found before the last concept (*drops of molten silver*) and no reinterpretation will be necessary. On the other hand, if there are many propositions matching the beginning of that sentence, it is possible that a reinterpretation occur. However, one can show that, under reasonable assumptions, the contents of the knowledge base does not affect the basic result that metaphor-first sentences are understood more slowly than metaphor-last sentences[1]. I prove this assertion only for the case of three-concept sentences.

Let us assume that f is the probability of finding the right interpretation (*The stars filled the night sky*) for the metaphor-last sentences after reading the first concept (*The night sky*) and that s is the probability of finding the right interpretation on the second concept (i.e., after reading both *The night sky* and *was filled*). We can safely assume that $f < s$: the more information you gather, the more likely you are to stumble on the right interpretation[2]. Let us also assume a certain probability r of rejecting a wrong interpretation. We are interested in estimating the expected number of interpretation switches for metaphor-last sentences. A switch can happen on the second concept (*filled*), or on the third concept (*drops of molten silver*), or on both. The probability of having only one switch on the second concept is $(1-f)rs+(1-f)r(1-s)(1-r)$: this sum corresponds to the case when a wrong interpretation is selected on the first concept, then it is rejected on the second and replaced either with the right one (first term) or with a wrong interpretation, which fails to be rejected on the third concept. The probability of having only one switch on the third concept is $(1-f)(1-r)r$ (I assume that, given all three concepts, the probability of finding the right interpretation is 1). The probability of switching on both the second and the third concepts is $(1-f)r(1-s)r$. Thus, the expected number of switches performed for a metaphor-last target is:

$$\begin{aligned} N_2 &= (1-f)rs+(1-f)r(1-s)(1-r)+(1-f)(1-r)r+2(1-f)r(1-s)r \\ &= (1-f)r(2-rs). \end{aligned}$$

For metaphor-first sentences, there is also a possibility of a switch on the second concept, if the interpretation selected on the first does not match it, or on the third concept, or on both. We can assume that the chance of selecting the right interpretation on the first concept is 0, as is the chance of selecting it on the second concept (i.e., you cannot guess a star interpretation after reading *Drops of molten silver* and *filled*). Then, if r is, as before, the probability of rejecting a wrong interpretation, there is a $r(1-r)$ chance of having a switch on the second concept only (that would mean that a wrong interpretation would be final). The probability of having a switch only on the third word is $(1-r)r$ and the probability of switching twice is r^2. Therefore, the expected number of switches for metaphor-first sentences is:

[1] However, the size of the latency difference depends on the structure of the knowledge base.

[2] But this assumption is not necessary for our proof.

$$
\begin{aligned}
N_1 &= r(1-r) + (1-r)r + 2r^2 \\
&= 2r.
\end{aligned}
$$

Then, we compute the difference in the number of switches between the two conditions:

$$
\begin{aligned}
N_1 - N_2 &= 2r - (1-f)r(2-rs) \\
&= r(2f + rs(1-f)) \\
&\geq 0, \text{ because } 0 \leq f, r, s \leq 1
\end{aligned}
$$

Therefore $N_1 > N_2$. We have shown that the expected number of switches is higher for metaphor-first sentences than for metaphor-last sentences and, therefore, the model takes longer to process the former[3]. The results in Table 4.2 correspond to $f = 0.36$, $s = 0.53$, and $r = 0.81$.

This demonstration is actually pessimistic, because it assumes equal cost of switches for metaphor-first and metaphor-last targets. In fact, for the latter, switches take less time because of interpretation priming: the old candidate interpretation (which was just rejected) helps the selection of related interpretations (see Section 3.2). For metaphor-last sentences, although initial interpretations may be wrong (e.g., after reading *The night sky was filled*, a possible candidate proposition is *The night sky was filled with airplanes*), in most cases they are more related to the correct interpretation than the bad candidates for metaphor-first sentences. For instance, suppose that the candidate interpretation after reading *Drops of molten silver filled* is *Drops of molten silver filled the bowl*; the bowl interpretation and the correct stars interpretation (*The night sky was filled with stars*) are less similar than the airplane interpretation and the stars interpretation. Thus, less activation spreads from the goal in the case of metaphor-first sentences and the interpretation switch is more expensive than for metaphor-last sentences.

A critical assumption made by the model is that the topic and the vehicle of the metaphor (e.g., *drops of molten silver* and *stars*) are semantically similar. The similarity should be high enough to ensure that, once all concepts are available, the model will find the right interpretation[4]. The value of this similarity can influence the latency and the model's ability to find the right interpretation. The results reported in Table 4.2 are obtained with the similarity between topic and vehicle set to 0.70. Table 4.3 shows the predictions of the model for other similarity settings. Decreasing the similarity to 0.50 does not affect the model too much: the latencies on the correct trials (i.e., on trials in which the

[3]Note that the switch time does not depend on the position of the word on which it occurs.

[4]Remember that we assumed that the probability of finding the right interpretation after reading the last word is 1.

		Similarity				
		0.25	0.35	0.45	0.50	0.70
Latency correct	metaphor-first	4.54	4.46	4.41	4.40	4.30
	metaphor-last	4.38	3.90	3.88	3.80	3.68
Latency all	metaphor-first	5.14	4.44	4.40	4.39	4.29
	metaphor-last	4.38	3.90	3.88	3.79	3.68
Error rate	metaphor-first	.792	.004	.020	.010	.006
	metaphor-last	.342	.014	.000	.002	.000

Table 4.3: Variation of model predictions with similarities between the vehicle and target of the metaphor. *Latency correct* stands for average latencies on targets for which the model is able to find the correct interpretation. *Latency all* stands for the average latency on all targets.

model finds a correct interpretation) suffer a slight increase and the model finds a wrong interpretation or no interpretation for about 1 percent of the metaphor-first sentences. This trend is accentuated when the similarity drops at 0.45 or 0.35, but the difference between the latencies for metaphor-first and metaphor-last targets is maintained around 0.5 s to 0.6 s. For similarity values of 0.25, the error rate increases dramatically: many sentences are not comprehended correctly and there is a smaller latency difference (0.16 s) between the two conditions for correct trials.

Thus, the model predicts that there is a threshold of similarity between the vehicle and the target, under which correct comprehension of metaphoric sentences is rare and takes a long time. It is not clear to what extent the value of the similarity maps onto the goodness of a metaphor; however, Blasko and Connine (1993) proved that metaphor goodness can facilitate comprehension for unfamiliar metaphors. On the other hand, in the study discussed in this section, Gerrig and Healy varied the quality of the metaphor (e.g., *The night sky was filled with drops of molten resin*) and obtained no significant difference in latencies between the comprehension of good and bad metaphors, although the bad-metaphor targets tended to take longer. The position effect was valid for both types of metaphors, separately or considered together. It is possible that people judge as unacceptable a metaphor with a vehicle–topic similarity under the threshold, so the metaphor goodness studies may actually look only at metaphors over this threshold.

Table 4.4 lists the ACT-R parameters that were estimated for this model to obtain the exact latency results. The high reading time for a word (actually for a noun phrase or verb phrase, as the model does treat heads and modifiers in one step) reflects the complexity of the phrases involved. The probability to stop searching corresponds to the probability of firing the production *stop-search* in Table 3.3: when the model does not find an interpretation matching the current word, it has the options to either search again or stop and go

Parameter	Abbreviation	Value	
Word reading time (s)	R	0.40	
Latency factor (s)	F	0.28	(see Latency Equation 2.4)
Activation noise	ans	0.24	
Retrieval threshold	rt	-0.55	
Probability to stop searching	stop	0.38	

Table 4.4: Parameters used by the ACT-R model in the Gerrig and Healy's (1983) task.

How many animals of each type did Noah take on the ark?
In the biblical story, what was Joshua swallowed by?
What is the nationality of Thomas Edison, inventor of the telephone?
In the novel "Moby Dick," what was the color of the whale that Captain Nemo was after?

Table 4.5: Stimuli used in the original Moses-illusion study (Erickson & Mattson, 1981).

to the next word.

4.2 Moses Illusion

Erickson and Mattson (1981) were the first who studied Moses (or semantic) illusions. They asked their subjects to look for distortions in sentences such as *How many animals of each kind did Moses take on the ark?* Surprisingly, people failed to find the distortions in such questions, in spite of knowing the corresponding undistorted facts (e.g., that Noah, rather than Moses, took the animals on the ark). As a dependent measure in their study, Erickson and Mattson defined the **illusion rate** as the percentage of failures to report distortions out of cases in which the correct answer is known. Thus, the illusion rate is based on the number of subjects who have the correct knowledge, rather than on the total number of participants.

Table 4.5 shows the four questions used as stimuli in the Erickson and Mattson's experiment. Even though the Moses-illusion effect was present for all items (the illusion rate was over 40 percent), people tended to fall most often for the Moses question (at an illusion rate of about 81 percent).

Whereas Erickson and Mattson's study drew attention to Moses illusion, it had several methodological shortcomings — for example, the small number of items used and the lack of a control condition, in which subjects would respond to the undistorted variants of the same questions. Other more rigorous studies followed up; one of them (Reder & Kusbit, 1991) also introduced a slightly different paradigm, the **gist task**. Unlike for the original Erickson and Mattson's task (henceforth called **literal task**), in which subjects had to detect distortions in Moses-illusion type of questions, in the gist task they needed to ignore the distortions and answer the questions as if they were undistorted. For example, the correct answer to the Moses question is *distorted* in the literal task and *two* in the gist task. Whereas for the literal task, the illusion rate is the dependent variable of choice, for the gist task the corresponding measure is the percentage of correct answers. Note that the gist task resembles metaphor comprehension: to respond correctly, subjects need to ignore the literal meaning of the sentence and process only those features of the distortion (or metaphoric word) that are relevant for the current context.

The next section presents two experiments that followed up on the original Moses-illusion study: Reder and Kusbit (1991) and Ayers et al. (1996). Section 4.2.2 shows how the sentence-processing model accounts for subjects' behavior in both literal and gist tasks. Specifically, the model simulates latency data from one of the experiments by Reder and Kusbit (1991) and illusion rates and percentages correct, as reported by Ayers et al. (1996).

4.2.1 Behavioral Data for Moses Illusion

Reder and Kusbit (1991) report several experiments intended to make subjects more sensitive to distortions. In this dissertation, I only look at latency data from Experiment 1[5] (Table 4.6). This experiment compared latencies for correctly answering distorted questions (e.g., *How many animals of each kind did Moses take on the ark?*) with those for answering undistorted questions (e.g., *How many animals of each kind did Noah take on the ark?*). Whereas in both gist and literal tasks there was no statistically significant difference in latency between the distorted and undistorted questions, subjects responded faster in the gist task than in the literal task. Also, in the gist condition, they tended to take longer (but not significantly longer) to answer correctly the distorted questions than to respond to the undistorted questions. These results indicate that in the literal condition people process more carefully the questions than they do in the gist condition.

Whereas, generally, people find the literal Moses-illusion task difficult, they rarely fall for certain distorted questions, such as *Who was the first man who walked on the sun?* Even the first Moses-illusion experiment (Erickson & Mattson, 1981) showed that not all distortions are equally good at tricking people: the illusion rate was much higher for the Moses question than for the other three questions in Table 4.5. Ayers et al. (1996) compared illusion rates for good and bad distortions embedded in similar sentences. They looked at three variants of the same question: one containing a good distortion, one containing a bad

[5] Reder and Kusbit's original experiment had two conditions: one in which the questions contained four to six terms associated to the answer and the other in which the questions had two or three associated terms. Here I present latency data aggregated over the two conditions.

	Data		Model	
Question	Literal	Gist	Literal	Gist
undistorted	4.25	3.69	4.27	3.04
distorted	4.29	3.88	4.23	3.78

Table 4.6: Mean response latencies (s) for correct responses in the gist and literal tasks for semantic illusions: data and model. (The data are adapted from Experiment 1 in Reder and Kusbit, 1991.)

	Illusion rate (literal)		Percentage correct (gist)	
Question	Data	Model	Data	Model
undistorted	7	3	82	90
good-distortion	46	51	76	87
bad-distortion	29	26	74	73

Table 4.7: The illusion rates in the literal Moses-illusion task and the percentage correct in the gist Moses-illusion task: data and model. (The data are adapted from Ayers, Reder & Anderson, 1996).

distortion and one containing the undistorted term. For example, the three variants could be *How many animals of each kind did Moses take on the ark?* (good distortion), *How many animals of each kind did Adam take on the ark?* (bad distortion) and *How many animals of each kind did Noah take on the ark?* (undistorted term). Ayers et al.'s (1996) results showed that people fell more often for the good-distortion questions than for the bad-distortion ones (Table 4.7).

Two studies by van Oostendorp and colleagues (van Oostendorp & de Mul, 1990; van Oostendorp & Kok, 1990) attempted to understand what makes a good distortion. They constructed triplets comprising a good distortion, a bad distortion, and an undistorted term and asked subjects to generate attributes for each of them and for the corresponding context frame. For instance, subjects had to generate attributes for *Moses, Adam,* and *Noah* and also for the missing term in the sentence *...took two animals of each kind on the ark.* Based on those attributes, van Oostendorp and colleagues computed the semantic overlaps between the undistorted terms and the good and bad distortions, respectively; they also computed the overlap between the distortions and the context frame. Their results showed that the overlap with undistorted terms and context frames was significantly lower for bad distortions than for good distortions. Moreover, as in the Ayers et al.'s (1996) experiment, the illusion rate was higher for good distortions than for bad distortions.

4.2.2 Simulation of Moses-Illusion

In accounting for the data from Moses-illusion experiment, my model starts from the findings of van Oostendorp and colleagues (van Oostendorp & de Mul, 1990; van Oostendorp & Kok, 1990). The model assumes that, for the illusion to work, there should be a significant semantic overlap between the distortion and the context frame (i.e., between *Moses* and the missing agent in *... took two animals of each kind on the ark*). Consequently, the difference between the good and bad distortions is the semantic similarity with the context frame: the smaller the similarity, the worse the distortion.

Remember from Section 3.2 that, on each new concept, the model attempts to either find an interpretation or validate the current one. If the model ends the processing of one concept with no valid interpretation, it produces a bug, recording that comprehension failure. In the literal task, the model considers a sentence distorted if it was not able to find an interpretation for it or if it produced one bug while comprehending it. Conversely, the model falls for a distorted sentence if its final interpretation is the same as if the sentence were undistorted and if it has formed no bug. In the gist task, the bugs are ignored: the answer is considered correct if the model found the right interpretation and incorrect if it found a wrong interpretation or none.

Just before reading a distortion, the model's candidate interpretation can be either "undistorted" (i.e., that of the corresponding undistorted sentence) or wrong[6]. If the model chose the "undistorted" interpretation, then, depending on the similarity to the context frame, the distortion may validate the interpretation (Figure 4.2a) or not (Figure 4.2b); failure of validation results in a bug and therefore in a *distorted* answer. Thus, the higher the semantic similarity between the distortion and the undistorted term, the lower the chance of a bug. On the other hand, if the candidate interpretation is wrong, it will be rejected; whether it will be replaced by the "undistorted" interpretation depends on the amount of converging activation from the distortion and from the previous two concepts (parts c and d in Figure 4.2). For good distortions, which are highly similar to the context frame, the activation spreading may be enough to select the "undistorted" interpretation. Therefore, the cases such as those depicted in parts b and d of Figure 4.2 (in which a bug is formed on the second concept) are more frequent for bad distortions than for good distortions.

[6] A third possibility is the model having no candidate interpretation. In that case, if the distortion is not the first word, then the model has formed a bug corresponding to that failure of finding an interpretation and, therefore, responds *distorted* in the literal task regardless of how it processes the distortion.

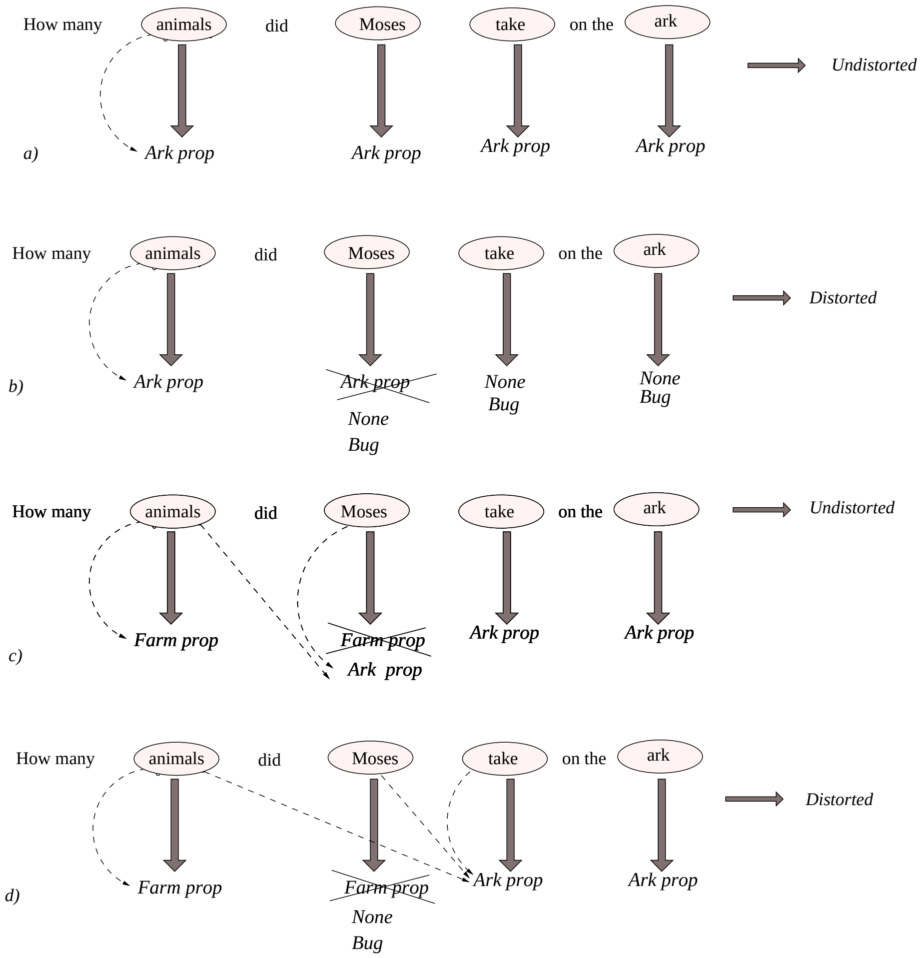


Figure 4.2: Model's processing of distorted questions in the Moses-illusion task.

Note that the model makes the prediction that distortions at the end of the sentence are ignored more easily than those at the beginning. If the distortion is the first or second content word in the sentence, there may not be enough spreading activation to select the "undistorted" proposition. Thus, for the question *How many animals did Moses take on the ark?*, if *animals* does not appear in any Moses contexts, there is a relatively high chance to end up with no interpretation on the word *Moses* (see part b in Figure 4.2) and produce a bug, which would lead to detecting the distortion. If *Moses* occurred later in the sentence, the activation from the other words may be enough to select the "undistorted" interpretation; then the similarity between *Moses* and *Noah* may be high enough to pass the matching test[7]. More experimental data needs to be collected to confirm the influence

[7] This argument depends on the reasonable assumption that it is easier for the distortion to pass a match

of the distortion position on the illusion rate.[8].

To summarize the behavior of the model on distorted sentences, the higher the similarity between the distorted term and the context frame, the more likely the model is to fall for the distorted sentence. Consequently, the model predicts that the illusion rates for good-distortion questions are greater than those for bad-distortion questions. The results of the simulations are shown in Table 4.7.

The model behaves essentially in the same way in both the literal and the gist task, but it treats bugs differently: for the literal task it uses them to give the *distorted* answer, for the gist task it ignores them. Another difference is the value of the retrieval threshold: for the gist task, the retrieval threshold is lower than for the literal task. A low retrieval threshold reflects tolerance for nonmatching words — although the negative activation spreading from those words decreases the activation of the "undistorted" interpretation (i.e., of the interpretation of the undistorted sentence), the latter may still be over the retrieval threshold. Thus, the low retrieval threshold allows the selection of the "undistorted" interpretation even for bad-distortion questions.

In most Moses-illusion studies, subjects were instructed to respond as soon as possible. To simulate this type of time pressure, the model is eager to give as soon as possible the *distorted* answer in the literal task and the *undistorted* answer in the gist task. Therefore, the model has a certain probability of stopping before the end of the sentence, if it produced a bug in the literal task and if it found an interpretation in the gist task[9]. Consequently, the model tends to stop before the end of the sentence on distorted sentences in the literal task and on undistorted sentences in the gist task.

With respect to latency predictions, the effective sentence-processing times for distorted sentences that are detected as distorted are longer than the corresponding times for undistorted sentences, because for the former there are, on average, more retrieval failures. However, for the undistorted questions in the literal task, the model spends extra time to perform integration at the end of the sentence; also the model never stops before the end of the sentence for these questions, because it has to make sure that the words yet to come will not produce a bug. These countervailing factors lead to approximately equal latencies for distorted and undistorted sentences in the literal task, as shown in Table 4.6.

---

test than to select the "undistorted" interpretation by itself or with one other word.

[8] Jaarsveld, Dijkstra, and Hermans (1997) manipulated the position of the distortion in the sentence and did not obtain a significant effect on the illusion rate. They used a verification task, in which Dutch subjects judged either an active sentence such as *Moses took two animals of each kind on the ark* or its passive equivalent (*Two animals of each kind were taken on the ark by Moses*). However, this study did not report baseline percentages of correct answers for undistorted sentences and some hidden effects may have taken place. For instance, subjects may have had a bias to reject passive sentences (corresponding to final-position distortions) and the potential position effect might have been canceled by this bias. Further investigation of this issue is needed.

[9] Ayers et al. (1996) found that subjects also have the same tendency of stopping before the end of the sentence.

In the gist task, there are two reasons why the model tends to be faster for undistorted sentences than for distorted sentences (see Table 4.6): it stops earlier for the former and retrieval failures on the distorted term (which take longer than successful retrievals) may occur for the second[10]. Note that, unless the model stops before the end of sentence, the integration cost in the gist task is paid by all types of questions.

The difference between the gist and the literal task in the model is that, for the gist task, there are fewer retrieval failures than for the literal task, because of the lower retrieval threshold. Retrieval failures are costly not only because they take more time than successful retrievals[11], but also because they imply extra processing for finding another interpretation.

Unlike for the metaphor-position simulation, the predictions of the model for the Moses-illusion task do not depend on the contents of the knowledge base. The number of propositions that may overlap with the correct interpretation makes the task equally hard for the distorted or undistorted questions. The only difference may rise from the number of contexts involving the distortion (e.g., *Moses parted the Red Sea*) that overlap with the right interpretation. However, if the overlap if significant, it may also affect the undistorted sentences — deciding which is a distorted sentence of what thus becomes a more difficult task.

The results presented in Tables 4.6 and 4.7 were obtained by assuming that the semantic similarity between the good distortions and the undistorted terms (appearing in the context frames) was 0.38 and the similarity between the bad distortions and the undistorted terms was 0.28. As expected, the illusion rate in the literal task and the percentage of correct answers in the gist task are monotonically increasing functions of the similarity between the distortion and the undistorted term. Table 4.8 shows how the performance of the model varies for different similarity values between the distortion and the undistorted term. As expected, the illusion rate in the literal task and the percentage correct in the gist task both increase with similarity. Also, the less similar the distortion is to the undistorted term, the longer it takes for the model to answer in both tasks. The increase in latency is caused by long retrieval times for dissimilar distortions; in the gist task, it is also caused by the extra processing for finding an interpretation.

Table 4.9 lists the values of the other ACT-R parameters that were used to produce the results in Tables 4.6 and 4.7. The base level and the probability to stop searching were kept at the same values as in the simulation for Gerrig and Healy's (1983) experiment.

[10] These failures to retrieve are not fatal for the gist task, because bugs are not taken into consideration.

[11] Note that a retrieval failure in the gist task cannot be compared with a successful retrieval in the literal task, because the model uses different values for the retrieval threshold in the two tasks.

Task	Measure	Similarity 0.18	0.28	0.38	0.48
Literal	Illusion rate	0.20	0.31	0.50	0.67
	Latency (s)	4.53	4.32	4.21	4.04
Gist	Accuracy	0.51	0.70	0.87	0.92
	Latency (s)	4.46	4.10	3.73	3.71

Table 4.8: Performance of the model as a function of the similarity between the distorted and the undistorted terms.

Parameter	Abbreviation	Value	
word reading time (s)	R	0.60	
latency factor (s)	F	0.03	(see Latency Equation 2.4)
activation noise	ans	0.40	
retrieval threshold	rt	-2.30	for the literal task
		-2.60	for the gist task
probability to stop searching	stop	0.38	
probability to stop before end	p_s	0.02	for the literal task
		0.50	for the gist task

Table 4.9: ACT-R parameters for simulation of Moses illusion.

Chapter 5

Sentences in Context

Chapter 3 described the basic mechanism underlying the sentence-processing model. Chapter 4 applied this model to two tasks: position effects on metaphor understanding and Moses illusion. Whereas the first task was an instance of sentence comprehension, as was the Moses-illusion gist task, the Moses-illusion literal task exemplified sentence verification. In this chapter, I show how the sentence-processing model extends to sentences embedded in discourse. In particular, I discuss how the model makes truth judgements of sentences based on a text (Section 5.1) and how it is able to relate novel sentences to preceding discourse (Section 5.2). All further discussion remains valid whether or not discourse is present (i.e., it also applies to isolated-sentence verification and to reading novel isolated sentences). The only distinction between text processing and isolated-sentence processing in this model is the prior knowledge: whereas in the case of isolated sentences, the knowledge base is the long-term memory, for text processing, the knowledge base comprises only propositions stated in the preceding context. Ignoring background knowledge for discourse processing is a simplification intended to avoid the issue of how episodic text memory relates to long term memory. Although the relationship between background and text knowledge is an important one, scrutinizing it is beyond the scope of this dissertation.

5.1 Sentence Verification

Sentence verification refers to judging whether or not a probe sentence is true, based on existing knowledge. The existing knowledge may comprise either long-term–memory facts or preceding-discourse propositions; I focus only on the latter case, although this discussion holds for verifying isolated sentences.

In Section 3.2 we saw that the final product of comprehension is an interpretation that matches best the input sentence. In some cases, the model may not be able to find such an interpretation. The lack of a final interpretation for a sentence suggests that

With the help of her godmother, Cinderella was able to go to the prince's ball dressed like a queen. Her stepmother became green with envy when she saw how beautiful Cinderella looked and she hated Cinderella even more. Cinderella was so charming that the prince danced all night only with her. When the clock started to strike midnight, she remembered that she had to go home; in her flight, she lost one slipper.

Table 5.1: Cinderella passage.

such an interpretation may not exist and that the sentence is either false or conveys new information. However, in a verification task, the only "true" probe sentences are those that are not novel[1]; if the model does not find any interpretation for a sentence, it will consider it false.

Figure 5.1(a) exemplifies the situation when the model fails to find an interpretation at the end of a probe sentence. Suppose that the model read the Cinderella passage in Table 5.1, acquired a propositional representation of that passage, and, then, had to judge the probe *Cinderella played the piano.* After processing this sentence, the model ends up with no interpretation, because, although there are more propositions about Cinderella in the story, none of them matches the input. Therefore, the model judges the sentence as *false*. Note that, during the process of searching for an interpretation, each time the model fails to find a candidate interpretation, it creates a bug (see Section 3.2), which records the context (word, thematic role, previous interpretation, etc.) at that moment. These bugs represent memories of comprehension failures; they also may be used for judging a sentence as false, as in the Moses-illusion literal task. To see why a final interpretation cannot be the sole criterion for judging the truth of a sentence, let us look at the example in Figure 5.1(b): at the end of the probe *The stepmother lost the slipper*, the model produced an interpretation for the sentence, even though that interpretation did not match all the previous words. Because the model is content with the highest-activation proposition matching the current word, if the positive activations from *lost* and *slipper* counterbalance the negative activation from *stepmother*, the proposition *Cinderella lost her slipper* can be retrieved and accepted as an interpretation. If the existence of a final interpretation were the unique criterion for truth judgement, the sentence *The stepmother lost her slipper* should be judged *true*. However, the existence of a bug recording that the model found no interpretation when it read the word *lost*, shows that there was some inconsistency between *lost* and the previous words. Therefore, retrieval of a bug prevents the model from judging the sentence as true, in spite of having found a final interpretation for it.

[1]The model does not deal with plausibility judgements, but with establishing truth based on facts explicitly stated in the prior context. Thus, the truth judgement is actually a *studied/not studied* judgement.

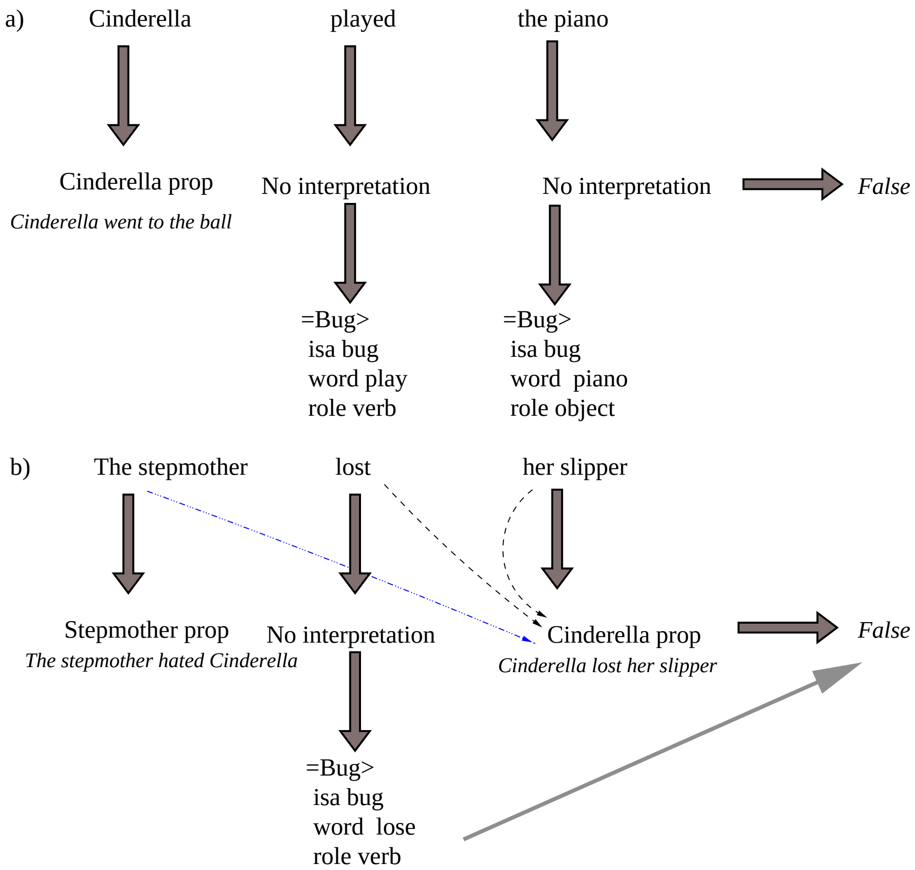


Figure 5.1: Verification of false probe sentences based on preceding Cinderella passage (see Table 5.1). a. Comprehension ends with no interpretation for the sentence and with two bugs. b. Comprehension ends with an interpretation and with a bug.

This procedure for sentence verification says that, each time a failure of finding any matching interpretation happens, that failure event is recorded and forms the basis for judging the truth of a sentence. Note that the lack of a final interpretation is a particular case of such a failure event: if the model ends with no interpretation, there is a bug that keeps track of that failure.

To summarize, in a verification task, the model considers a sentence true if it found a final interpretation and if it generated no bugs during its comprehension. The model judges a sentence as false if it is able to retrieve a bug produced while processing that sentence.

5.2 Novel-Sentence Comprehension

Our discussion of sentence comprehension was limited to the case when there was a proposition (be it from long-term memory or from discourse) corresponding to the input sentence; that proposition formed the meaning (interpretation) of the input. However, a lot of input sentences are novel and do not have correspondents in prior knowledge: only think how many new facts people learn from birth!

In the previous section we have seen that sometimes the model finds an interpretation that does not match perfectly all the words in the sentence, but at least matches some of them. Even though such an incomplete match is not acceptable for sentence verification, it may provide the basis for comprehension and may allow the model to relate new information to old knowledge. The given–new theory, proposed by Haviland and Clark (1974), argues that sentences contain **new** and old, **given** information and that the new information is understood in terms of the old one. Haviland and Clark suggested that people first find antecedents in memory for the given part of a sentence and then attach the new information to those antecedents.

Remember that my model attempts to find a candidate interpretation for the sentence as soon as possible. Even though such candidate interpretations may be invalidated and replaced by others further on, they can have a final word in the comprehension. Candidate interpretations are partial matchings to the input. If, at the end of the sentence, no final interpretation is found, one of these partial matchings may be retrieved and used as a point of attachment for the new information. Thus, old candidate interpretations serve as anchor points or **hooks** into the discourse. Hooks serve the role of linking a novel sentence to existing knowledge. They are treated as regular, matching interpretations are; only the existence of a bug (in the case of hooks) discriminates between the two. Whereas the model can reject several candidate interpretations, only one of them is used as a hook. The natural choice is the candidate that was rejected last, because that interpretation matches presumably more concepts from the sentence than any other previously rejected candidate. After the hook was selected, it is upgraded to the status of interpretation and, as for sentences with a regular interpretation, the model updates each of the propositional links corresponding to the input sentence to point consistently to the chosen hook.

Suppose that the model read the passage in Table 5.2 and now it has to read the next sentence, *In the forest, Snow White found a small house.* Figure 5.2(a) shows how the model processes that input sentence. Two propositions from the prior text are considered as candidate interpretations (they are the animals proposition and the Snow-White proposition, on the first and, respectively, the second word); unfortunately none is validated by further words, so the model fails to find a final interpretation for the sentence. However, to relate the current sentence to the passage, the model retrieves the last invalid candidate interpretation, *Snow White was lost in the forest,* and uses it as a hook. As Figure 5.2(b) indicates, all the proposition links in the final representation of the input sentence (see

When Snow White's stepmother found out that Snow White was more beautiful than she was, she ordered a hunter to take her into the woods and kill her. The hunter felt pity for the young girl, so he did not kill her, but let her free into the forest and told her to beware of her stepmother. Soon, Snow White was lost in the forest and became very scared. Many animals lived in the forest and some were not friendly.

Table 5.2: Snow-White passage.

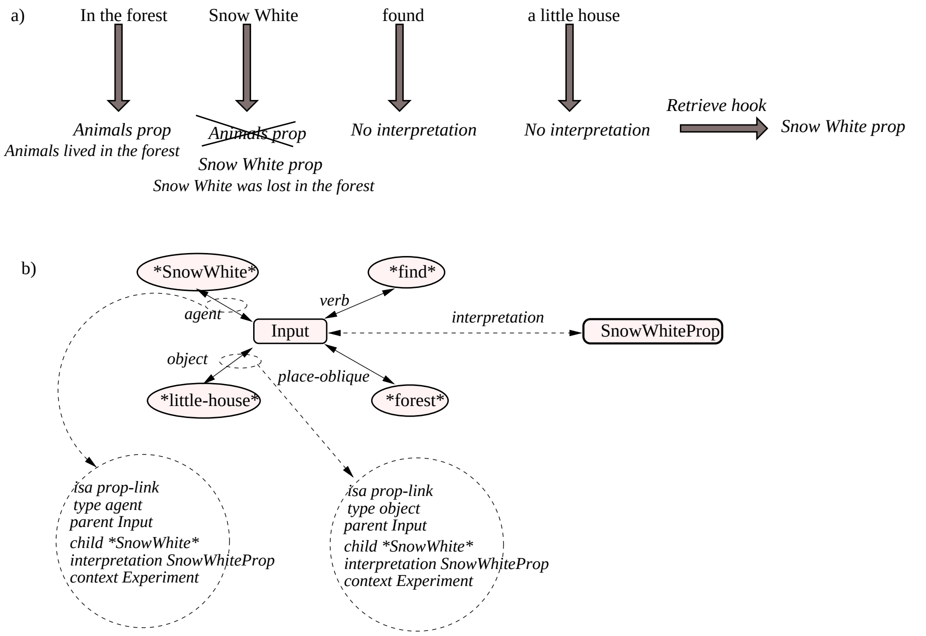


Figure 5.2: Comprehension of a novel sentence. a. Candidate interpretations. b. Final representation for the input sentence.

Section 3.1.2) have the filler of the slot *interpretation* updated to point to the Snow-White proposition.

Note that the role played by the hook is that of an approximate interpretation, known not to match exactly the sentence, but which is the best match in the passage. However, the model treats the hooks and the matching interpretations in the same way; to see why, let us compare the comprehension of two semantically equivalent sentences: *The stepmother lost the slipper* and *The slipper was lost by the stepmother.* First, let us go back to the active sentence *The stepmother lost the slipper*, which was used as an example in Figure 5.1(b); suppose that, instead of verifying this sentence, the model's task was

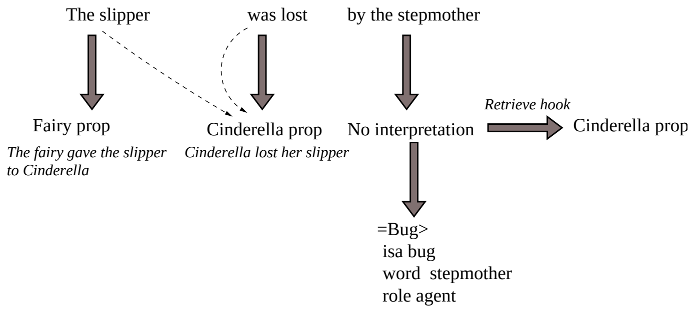


Figure 5.3: Comprehension of the sentence *The slipper was lost by the stepmother.*

to comprehend it, after reading the Cinderella passage in Table 5.1. The succession of candidate interpretations for comprehending this input sentence could be the same as the succession shown in Figure 5.1(b) and the final interpretation would be *Cinderella lost the slipper.* As we saw in Section 5.1, if the task were to verify the truth of the input sentence, the model would judge it as *false*, because of the bug formed on the verb. However, because the task is comprehension, the model does not pay attention to the bugs and accepts the proposition *Cinderella lost the slipper* as the final interpretation to the input. This solution makes sense if we think that, first, people do not perform extensive correctness checks during normal comprehension and, second, that it is in the interest of the cognitive system to relate the input to something already known as fast as possible. It also offers robustness to inherent noise and human errors. However, this interpretation does not match every word in the input sentence; in this respect, it is similar to a hook, which is a best partial match for the current input sentence. If the model needed to examine how good a match the final interpretation was, then it would be able to do so by checking for bugs.

Let us now examine the equivalent passive sentence, *The slipper was lost by the stepmother.* Despite the semantic identity with the sentence in Figure 5.1(b), if the model were to comprehend this sentence (Figure 5.3), it would end up with no interpretation, because the Cinderella proposition, although chosen as a candidate on the second word, would not match the last word and would be rejected. However, the model can retrieve a past interpretation of this sentence (the Cinderella proposition) and use it as a hook into the discourse. Thus, because of hooks, the model finds the same final interpretation for both the active and the passive variants of the same input *The stepmother lost the slipper.* If hooks and interpretations were different, then the two sentences would not have been semantically equivalent for this model. In other words, if the model can find interpretations that match the inputs only partially, there is no reason to treat hooks (which, by definition, are rejected interpretations which do not match perfectly the input) as different entities.

Chapter 6

Empirical Evaluation: Sentences in Context

In Chapter 5, we examined the behavior of the model on sentences embedded in context. In this chapter, we see how the discourse model fares on accounting for two empirical datasets (Budiu & Anderson, 2001, 2000a). Both studies were concerned with processing metaphoric sentences. The first study looked at the effect of metaphor familiarity on verification of metaphoric sentences. The second study compared the comprehension of sentences with various degrees of metaphoricity.

This chapter starts with an overview of the metaphor-comprehension literature. Section 6.2 presents the experiment in the metaphor-verification study (Budiu & Anderson, 2001) and the results of the simulations. Section 6.3 describes the metaphor-comprehension data (Budiu & Anderson, 2000a) and how the model accounts for them.

6.1 Metaphor Comprehension: Theory and Data

Perhaps the most famous and frequently refuted theory of metaphor comprehension is Searle's error–recovery theory (Searle, 1979). Searle claimed that, when confronted with a metaphor, people first try to understand the sentence literally and, in case of failure, they look for a metaphorical interpretation. The context dictates whether or not the literal meaning is appropriate. The recognition of a metaphor consists of three steps: first, a literal interpretation of the sentence is built; second, this interpretation is matched against the context; third, if no consistent matching can be found, a metaphorical interpretation is considered. A corollary of this theory is that people take longer to understand metaphorical utterances than to understand literal utterances, because they have to go through extra processing for the former.

Many psycholinguistic studies attempted to test this corollary. Among the first and most influential ones was Ortony et al.'s (1978). Ortony et al. (1978) showed subjects either a passage about a women's club meeting or about chickens on a farm and and followed each of them by a target sentence such as *The hens clucked noisily.* When it came after the first passage, the sentence had a metaphoric interpretation; after the second passage it was literal. Participants in Ortony et al.'s experiment read this sentence as fast in both conditions. Inhoff et al. (1984) replicated this study and obtained similar results. This result was interpreted as evidence that, when context is supportive, people process metaphoric sentences as fast as literal sentences and as a refutation of Searle's (1979) theory.

Not only do subjects sometimes access the meaning of a metaphor as fast as the literal meaning, but the metaphoric meaning can interfere with the literal meaning. Glucksberg, Glidea, and Bookin (1982) had subjects judge the literal truth of sentences of the form *A is B*. Subjects took longer to reject sentences that made sense metaphorically (e.g., *Some jobs are jails*) than to reject nonsense sentences (e.g., *Some apples are pears*), albeit both being literally false. Keysar (1989) extended Glucksberg et al.'s results in an ingenious experiment. He manipulated both the literal and metaphoric truth of sentences *A is B*, by varying the context that preceded them. He obtained shorter judgement times for the congruent conditions (in which literal truth matched metaphoric truth) than for the incongruent ones and interpreted this result as supporting the inseparability of metaphoric and literal processing. Keysar (1989) repeated the experiment and measured comprehension times; he obtained fastest reading times for sentences that were both literally and metaphorically true. The next fastest latencies were for sentences that were either literally or metaphorically true and slowest reading times for sentences that were both metaphorically and literally false.

The findings of Ortony et al. (1978), Inhoff et al. (1984), Glucksberg et al. (1982), Shinjo and Myers (1987), Keysar (1989) seemed to tip the balance in favor of the tenet that comprehension of literal and metaphoric sentences are governed by similar processes. However, later studies Janus and Bever (1985), Gibbs (1990), Onishi and Murphy (1993) undermined this view. Janus and Bever (1985) replicated Ortony et al.'s (1978) findings for metaphors embedded within a rich context; however, beside measuring the sentence-reading times, they looked at the reading times for the metaphoric nouns. Even though, like Ortony et al. (1978), they found no significant difference between reading times for metaphoric and literal sentences, the reading times for metaphoric nouns were longer than those for literal nouns. Janus and Bever (1985) interpreted his results as a refutation of the view that the same mechanism is involved in the comprehension of metaphoric and literal language.

A study by Gibbs (1990) also provided some support to the Searle's model of metaphor comprehension. Gibbs showed subjects short passages followed by either a metaphoric or a literal sentence. For instance, one such passage was about a boxing match and ended either with a metaphoric sentence such as "The creampuff did not show up for the match" or with its literal equivalent "The boxer did not show up for the match". Gibbs did find a reading time disadvantage for metaphoric sentences with respect to literals, but attributed this result to the type of metaphors used — anaphoric in his study versus predicative in those studies that had provided evidence for similar literal- and metaphor-comprehension processes (Inhoff et al., 1984; Shinjo & Myers, 1987; Glucksberg et al., 1982; Keysar, 1989). However, even though the metaphors employed in their study were also anaphoric, Ortony et al. (1978) failed to find a difference between sentence-reading times for literal and metaphoric sentences.

In conclusion, it seems that, even though Searle was wrong in his assumption that literal interpretation must always precede metaphoric interpretation, sometimes literal processing does precede metaphoric processing. In some experiments, on average, the metaphoric interpretation is available as fast as the literal one (Ortony et al., 1978) and in other experiments it is available later (Gibbs, 1990). Sometimes the metaphoric interpretation beats out the literal producing interference to the literal interpretation (Glucksberg et al., 1982). If the subjects can process the sentence with either the literal or metaphoric interpretation, they will be fastest (Keysar, 1989).

6.2 Metaphor Learning

In the previous section, we saw that, although in many cases people process metaphoric sentences as fast as they process literal sentences, this phenomenon is not universal (Gibbs, 1990; Onishi & Murphy, 1993). One hypothesis is that, as a metaphor gains familiarity, it comes to be processed as rapidly as the literal meaning. Indeed, there are a lot of words in natural languages that are such metaphorical extensions of old meanings, based on more or less obvious similarities between two concepts. Often an analogy gives rise to the same metaphor in several languages — as in the commonly met "*leg* of a table" or "*foot* of a hill", or even when using verbs such as the English *catch* or *grasp* to mean *understand* (Ullmann, 1966). Some of the new meanings of such words coexist with the old meanings, others replace them.

6.2.1 Behavioral Data (Budiu and Anderson, 1999)

The experiment by Budiu and Anderson (2001) showed that as people become more acquainted with a metaphor, they process it faster. In that study, Budiu and Anderson were concerned with learning new meanings for both metaphors and artificial words, which had no prior meaning. Here, I only report the results for metaphors, as they were obtained in Experiment 2 from Budiu and Anderson (2001). In this experiment subjects read a short passage and then judged the truth of a probe sentence. After each trial, subjects were given feedback about their answer. The probe could be either true or false and either metaphoric or literal. Metaphoric probes contained an anaphoric metaphor to denote a

Jim was a philosophy junior. In one of his classes, he noticed a very massive young man who was always sleeping and never paid any attention to the discussions. One day, somebody told Jim the man was a very good linebacker that had been all-state in football. So the mystery was solved: he was accepted at the university for his athlete rather than for his philosopher qualities.	Joe went to see the very famous wrestler, John Smith, in a match for the national title. John Smith was so big that he had the reputation that nobody could move him from where he was standing. And indeed, although the other wrestler was himself a massive man, he couldn't even make Smith budge an inch.
Metaphoric sentences:	
The bear was sleeping in the philosophy class. *[true]* The bear noticed a man sleeping in class. *[hard foil]*	The bear was competing for the national title.*[true]* The bear was competing for the school title.*[easy foil]*
Literal sentences:	
The athlete was sleeping in the philosophy class. *[true]* The athlete noticed a man sleeping in class. *[hard foil]*	The wrestler was competing for the national title.*[true]* The wrestler was competing for the school title[a].*[easy foil]*

[a]For the sake of clarity, this example has been slightly modified from the original experiment.

Table 6.1: Sample materials from Budiu and Anderson, 1999.

concept introduced in the passage. Thus, if the passage was about a bulky athlete who was very tired and always slept in class (first column in Table 6.1), a true metaphoric probe would be *The bear slept in class.* In this context, *bear* would (appropriately) refer to the bulky athlete. There were two types of false probes: easy and hard. The easy foils could be judged as false without understanding the metaphor. For instance, if the story was about a wrestler competing for the national title (see the second column in Table 6.1), *The bear competed for the school title* would be an easy foil because there was no character in the story who *competed for the school title.* In contrast, hard foils were foils that needed a correct identification of the metaphor referent in order to be judged false. For instance, in the sleepy-athlete story in Table 6.1, *The bear noticed a man sleeping in class* would be a hard foil because there is another character in the story (*Jim*) of whom the predicate *noticed a man sleeping in class* is true, so the predicate could not be used for

Block[a]		Data 1	Data 2	Data 3	Data 4	Model 1	Model 2	Model 3	Model 4
Trues	Met	53	83	82	84	59	72	77	80
	Lit	90	86	79	86	95	95	98	97
Easy foils	Met	100	85	72	89	92	87	88	87
	Lit	85	79	80	84	84	78	84	81
Hard foils	Met	73	73	70	72	76	75	75	76
	Lit	88	79	77	84	83	85	84	81

[a]Block numbers denote the order in which the blocks appeared in the experiment.

Table 6.2: Percentage of correct truth judgements in Budiu and Anderson (1999): data and model. (*Met* stands for *metaphoric*; *Lit* stands for *literal*.)

judging the probe correctly. To assess how metaphor comprehension changes with familiarity, Budiu and Anderson (2001) showed the participants the same metaphor eight times; the metaphor was embedded in different probes and preceded by different passages. Thus, there were eight passages about bulky athletes who were referred to as *bears* in the probes. (Table 6.1 shows two of these passages.) The authors report the accuracy and the time taken for truth judgements across different experimental blocks. Each experimental block contained 16 trials; in a block all metaphors were seen twice. The results are depicted in Tables 6.2 and 6.3. The accuracy was lower for true metaphoric probes at the beginning of the experiment, but it improved by the last block to become comparable with the accuracy for literals. Also, the latencies for metaphoric probes were longer in all conditions in the first blocks, but they were comparable with the literal latencies in the last blocks. These results, together with the definitions that subjects provided for the metaphoric words in a post-experiment test, indicated that, by the end of the experiment, new meanings were learned for the metaphoric words.

6.2.2 Simulation of metaphor learning

In the following discussion, one fundamental assumption is that subjects do learn new meanings for the metaphoric words, if they encounter them several times.

To account for the data reported by Budiu and Anderson (2001), the sentence-processing model described in Section 5.1 is embellished with a **word-learning** capability and with a **reevaluation** capability. Figure 6.1 exemplifies these two features for the case when the model must judge the sentence *The bear slept in class*, in the context of the sleepy-athlete story from the first column of Table 6.1.

Block[a]		Data				Model			
		1	2	3	4	1	2	3	4
Trues	Met	4504	3258	2889	2609	4350	3040	2870	2770
	Lit	3590	3103	2945	2846	3600	2720	26400	2600
Easy foils	Met	4177	3261	3357	2988	3550	2680	2650	2600
	Lit	4462	3072	3287	2931	3380	2730	2650	2570
Hard foils	Met	4411	3685	3463	3220	4430	3190	2950	2890
	Lit	3798	4266	3211	3258	3800	2970	2820	2840

[a]Block numbers denote the order in which the blocks appeared in the experiment.

Table 6.3: Latencies (ms) for the truth judgements in Budiu and Anderson (1999): data and model. (*Met* stands for *metaphoric*; *Lit* stands for *literal*.)

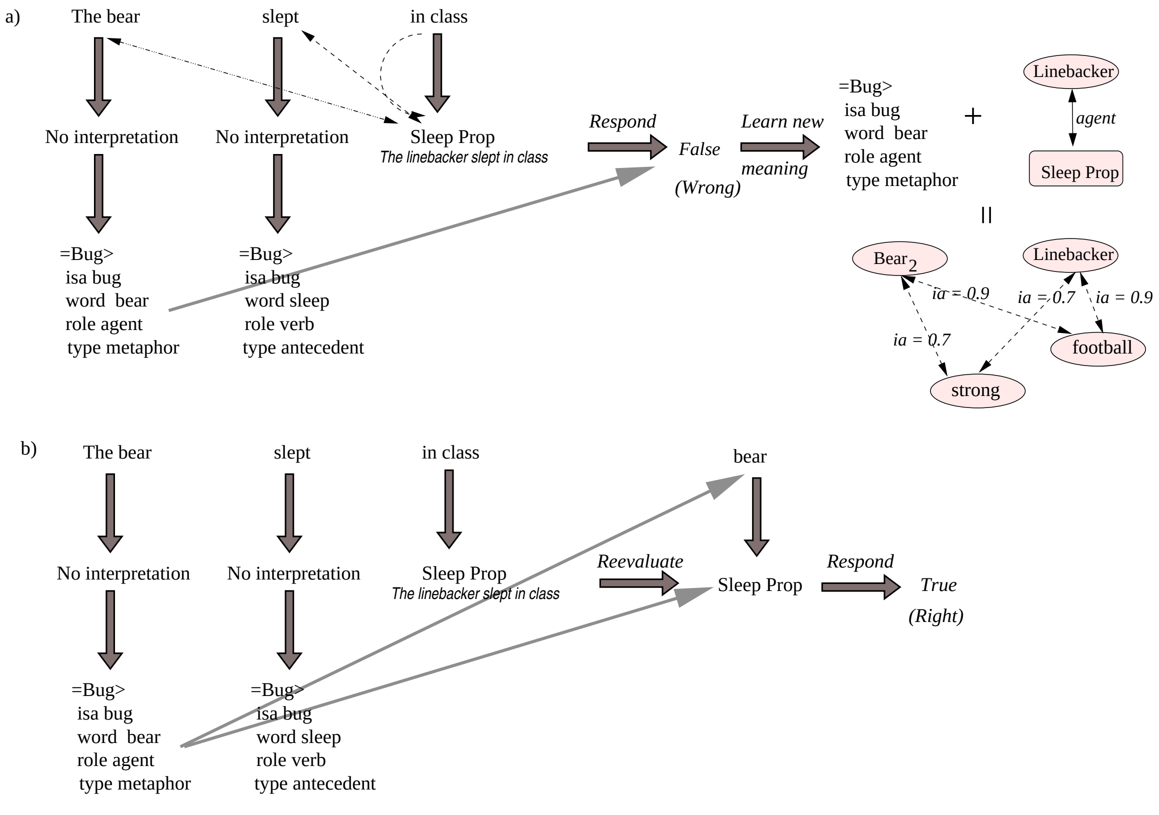


Figure 6.1: Processing true metaphoric sentences from Budiu and Anderson's (1999) experiment. a. Learning of a new meaning for a metaphor. b. Reevaluation of a metaphoric sentence.

Initially, the model tends to answer *false* to all metaphoric probes. Figure 6.1(a) shows that, when the model reads the word *bear*, it cannot find any interpretation matching it and, thus, it forms a bug. Although later on the model may find an interpretation (because of the additive effect of spreading activation from the other words in the sentence), the existence of the bug forces the model to answer *false* to a metaphoric probe (see Section 5.1). However, if it receives feedback that the correct response is *true*, it will create a new meaning for the word *bear* (on which the bug was generated), based on the final interpretation of the sentence. Thus, if, as in Figure 6.1(a), the final interpretation of the sentence *The bear slept in class* is the sleep proposition (*The linebacker slept in class*), then the model will check the corresponding role (henceforth called **seed**) in that interpretation — namely, *agent* — and assume that *bear* refers to that agent (i.e., to *linebacker*). Therefore, it will create a new meaning chunk for *bear*, which will inherit all associations of the seed *linebacker*.

One complication arises from the existence of multiple bugs. For instance, in Figure 6.1(a), two bugs were generated: one on the word *bear* and the other on the word *sleep*. The second bug appeared because the positive spreading activation from *slept* was not enough to countervail the negative activation spread from *bear*, so the model was not able to find any interpretation on the word *slept*. Given that it would be inappropriate to create a new meaning for *slept*, how should the model recognize the metaphor? The answer lies in the slot *type* of the bug: when it creates the new bug, the model checks whether the corresponding word has an antecedent in the context; if it does, then it will assign to it the type **antecedent**; otherwise, it will assign to it the type **metaphor**. The model creates new meanings only for bugs of type *metaphor*, because for these bugs (part of) the blame for the comprehension failure can be assigned for sure to the word on which they occur, whereas for antecedent bugs the sentence context, rather than the word itself, is inconsistent with the passage.

To summarize, initially, the model answers *false* to metaphoric sentences and learns new meanings only when it makes mistakes (i.e., on true metaphoric probes). However, if this were the only treatment of metaphors, then, when it saw a metaphor for the first time, the model would always answer *false* even to a true metaphoric sentence. The data show that subjects do sometimes answer *true* to probes containing a new metaphor. To make the model exhibit that behavior, I introduced an optional reevaluation phase, exemplified in Figure 6.1(b): when the model reaches the end of the sentence with an interpretation, before answering *false* when there is a metaphor bug, it may choose to reevaluate that bug. Reevaluation is roughly identical with the reprocessing of the word in the bug at the end of the sentence: it basically treats the bug word as an extra word in the sentence. The benefit obtained by reprocessing is that of matching instead of searching for an interpretation and is similar to the advantage of metaphor-last sentences in the model for the Gerrig and Healy's (1983) experiment: given that there is a final interpretation (i.e., sleep proposition in Figure 6.1b), if the bug word (i.e., *bear*) is similar enough to the corresponding role in that interpretation (i.e., *linebacker*), then it will be accepted. The similarity is judged (as is in regular matching — see Section 3.2) based on retrieval success or failure; however, reevaluation is more insistent than regular matching, in the sense that the retrieval is retried for several times if unsuccessful[1].

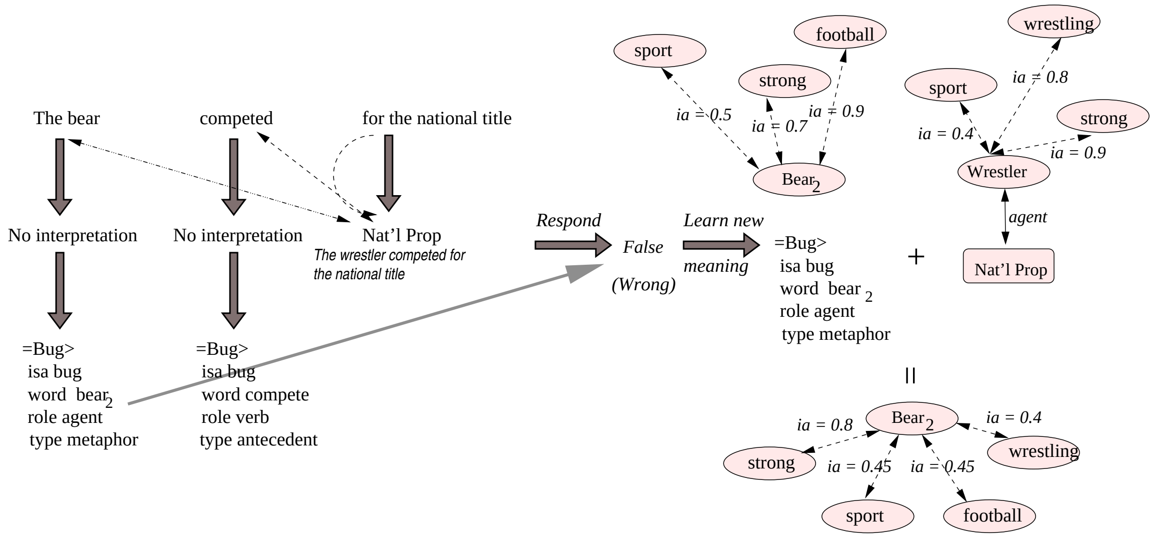


Figure 6.2: Updating the associations of a new meaning.

Once the model created a new meaning for the metaphoric word, it would retrieve it in all subsequent trials involving that metaphor and would treat it as a regular word. However, because the new meaning is virtually a copy of the original seed, it may be inappropriate in at least some of the further trials. For instance, the meaning *linebacker* for *bear* may be found inappropriate for the story about the wrestler (second column of Table 6.1). Thus, if the model responds *false* to a true metaphoric probe after it learned a new meaning, instead of creating yet another meaning (i.e., instead of creating a third meaning for *bear*, corresponding to *wrestler*), it will update the existent one by changing its associations to reflect the associations of the novel seed. The old meaning's associations get averaged with the associations of the new meaning (Figure 6.2). This mechanism of word learning allows the new meaning to get strong associations to only those concepts related to all seeds and weak association to those concepts that are related to only one of them.

I must acknowledge that such a model of word learning is not necessarily plausible. For instance, one legitimate question is how associations are computed and averaged; is such a process instantaneous as this model assumes? However, given that the focus of my model is sentence processing, rather than learning of new meanings, any mechanism that would perform some gradual word learning would be acceptable.

[1]A failed retrieval may succeed later on, due to fluctuations in the noise term added to the chunk activation — see Chapter 2.

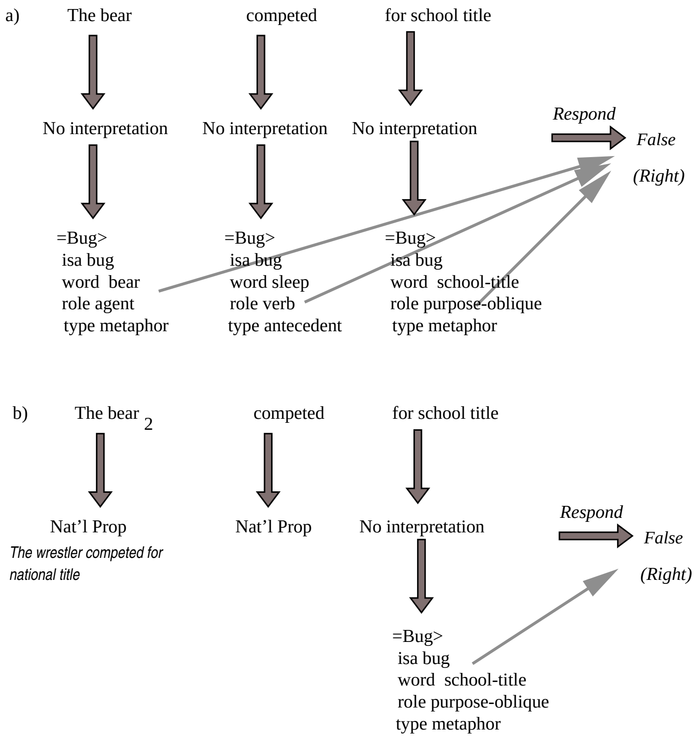


Figure 6.3: Model's processing of easy foils. a. The literal meaning of the metaphor is used. b. The new meaning of the metaphor is used.

We saw how the model judges metaphoric true sentences: initially, it answers *false*, unless it chooses to reevaluate the sentence. When the model answers *false* incorrectly, it creates a new meaning for the metaphor that is identical to the corresponding meaning in the context; the new meaning is subsequently used and refined on each error made on true probes involving that metaphor.

Let us now look at the behavior of the model on false probes. The model always answers *false* to easy foils, because there is no proposition in the context that matches them even partially; therefore, the model ends either with no interpretation for them or with at least a bug. Figure 6.3 shows how the model comprehends the easy-foil probe from the story in Table 6.1. Note that, for easy foils, it does not make a difference whether the traditional meaning or the new meaning of the metaphor is used, because in both cases a bug is produced (in Figure 6.3, it corresponds to *school title*) caused by the mismatch between the predicate and the context. However, if the mismatch is small, it may pass undetected and cause the model to answer *true*.

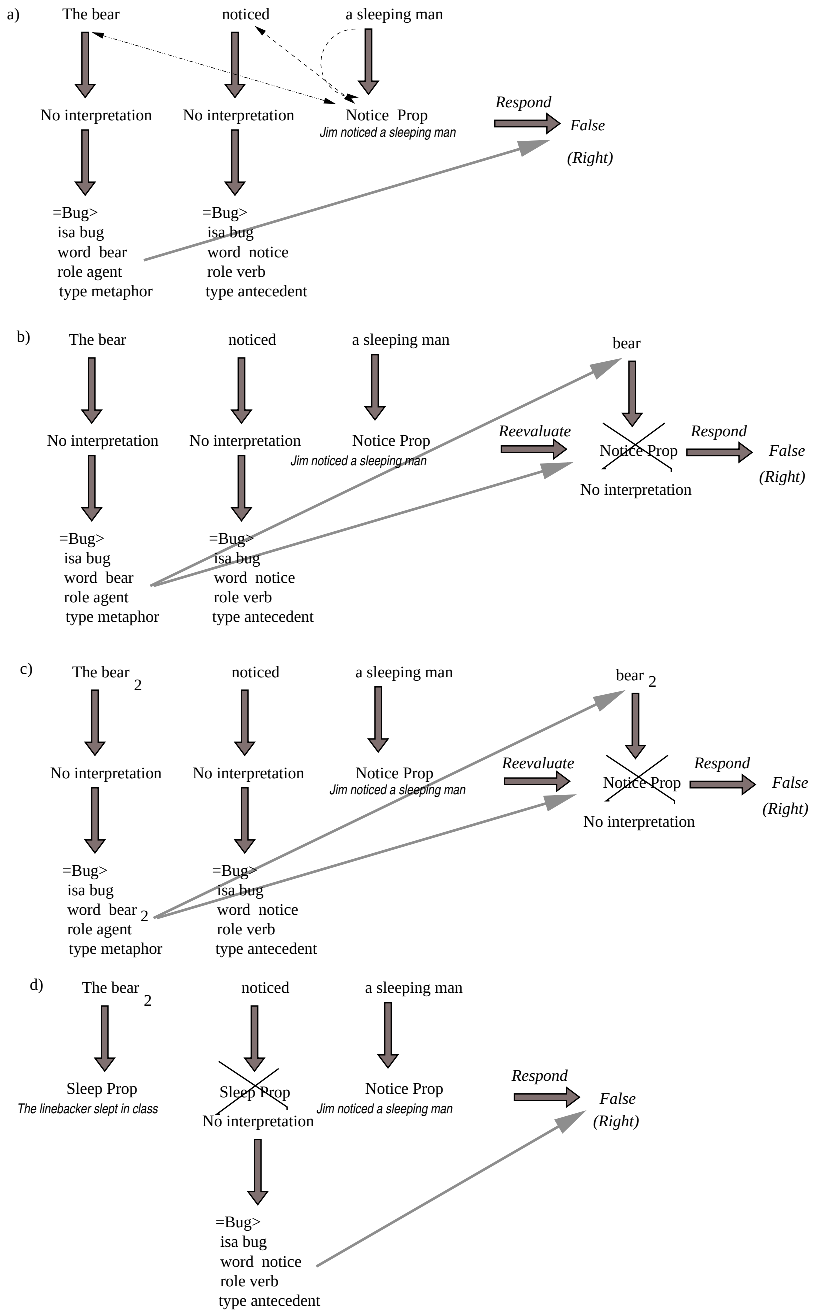


Figure 6.4: Model's processing of hard foils. a. The literal meaning of the metaphor is used. b. The literal meaning of the metaphor is used and reevaluated. c. The new meaning of the metaphor is used, but it does not match anything in the context. d. The new meaning has an antecedent in the context.

For hard-foil probes, the processing is more similar to the case of true metaphoric sentences. Figure 6.4 shows how the model comprehends the hard foil corresponding to the story from the first column of Table 6.1. Initially (see parts a and b of Figure 6.4), when there is no new meaning for the metaphor, the model produces a bug on the metaphoric word (i.e., *bear*), as for the other types of metaphoric sentences. However, because the hard foil contains a predicate that matches a proposition in the context, that proposition may be the final interpretation of the model (i.e., the notice proposition). Therefore, for a hard foil, as for a true metaphoric probe, the model has two options: either to answer *false* in virtue of the metaphor bug (Figure 6.4a) or to reevaluate (Figure 6.4b). Unlike for true sentences, the reevaluation rarely succeeds (unless the subject of the final interpretation is semantically similar to the metaphor). Therefore, whether or not the model chooses to reevaluate the metaphor does not make any difference with respect to the final answer, which is *false*. This answer is correct, so the model does not need to learn any new meaning for the metaphoric word.

If the model has already acquired a new meaning (parts c and d in Figure 6.4), the behavior depends on how well the new meaning was learned. Thus, if, as in Figure 6.4(c), the new meaning does not match the *linebacker* in the story, then its processing is identical with the case when the literal meaning of the metaphor is used (parts a and b of Figure 6.4). If, on the other hand, the new meaning has been captured to a greater degree (as in Figure 6.4d), then model may find an antecedent for it in the context (e.g., *linebacker* in the sleep proposition) and treat it as a literal word that matches that antecedent. However, the candidate interpretation matching the new meaning must be rejected on the second word (i.e., *notice*, which does not match the verb of the sleep proposition) and, next, a new interpretation, matching the predicate, may be selected (i.e., the notice proposition). The **antecedent** bug on the second word captures the inconsistency between the agent and the rest of the sentence, thus determining the answer *false*. Antecedent bugs are never reevaluated; they always offer a justification for judging a sentence false (even though they may be sometimes ignored if they occur together with metaphor bugs).

The processing of literal sentences follows the pattern described in Section 5.1. The results obtained by the model are given in Tables 6.2 and 6.3. Note that, according to the preceding discussion and to the considerations in Section 5.1, the model should be perfectly accurate for literal sentences and for false metaphoric sentence. However, subjects are not. The model accounts for errors in these cases by assuming occasional imperfections in materials: thus, for some stories, there may be a low similarity between the literal and the word that it denotes (e.g., *athlete* and *linebacker* for the story from the first column of Table 6.1), or a high similarity between a metaphor or literal and the story subject of a hard-foil predicate (e.g., *bear* or *athlete* and *Jim* in the same story), or a high similarity between an easy-foil predicate and one predicate occurring in the story (e.g., *competed for regional title* and *competed for national title*). Table 6.4 shows the values of various similarity parameters that were estimated for this model.

Similarity	Value	Scope	Example
vehicle–topic	0.20	all stories	between *bear* and *linebacker*
literal–topic	0.95	all stories	between *athlete* and *linebacker*
bad-literal–topic	0.35	one story	between *athlete* and *linebacker*
hard–vehicle	0.20	one story	between *Jim* and *bear*
hard–new-meaning	0.44	one story	between *Jim* and all main characters (e.g.,*John Smith*)
hard–literal	0.45	one story	between *Jim* and *linebacker*
easy–right	0.23	one story	between *school title* and *national title*
literal–literal	0.67	all stories	between *linebacker* and *John Smith*

Table 6.4: Similarities for simulation of the metaphor-learning task. The examples use the stories in Table 6.1.

The model captures the basic result that metaphoric sentences take longer than literal sentences in the beginning of the experiment, but behave as literal sentences with increased exposure to the metaphor. The model and the subjects are less accurate on true metaphoric sentences in the first experiment blocks. The accuracy of the model in the first block depends on the similarity between the metaphor and the literal denoted by it, on the probability of reevaluation, as well as on the insistence at reevaluation (i.e., on how many attempts to match the metaphor to the interpretation are made). Later, the model's increased accuracy on metaphoric true sentences reflects the acquisition of a new meaning, which becomes more accurate. A parameter that influences the speed of new-meaning acquisition and thus the accuracy on later blocks, is the semantic overlap between different potential seeds (thus, if the characters of all the stories have a lot in common, the meaning acquisition is smooth and fast). The latency difference between the metaphoric and literal trues in the first block is due mainly to reevaluation[2]. The insistence in reevaluation gives the size of that difference. The overall speedup between the first and the last experimental blocks is caused by production-strength learning (see Chapter 2) — namely, by practice with the task and, in particular, with those answer-reporting productions involving key presses.

Table 6.5 shows the values of the other ACT-R parameters involved in producing the results in Tables 6.2 and 6.3.

[2]It is also due to retrieval failure on the metaphor word — see Figure 6.1.

Parameter	Abbreviation	Value	
word reading time (s)	R	0.35	
latency factor (s)	F	0.10	(see Latency Equation 2.4)
activation noise	ans	0.38	
retrieval threshold	rt	-0.88	for the literal task
probability to stop searching	stop	0.38	
strength learning	sl	0.45	d in the Equation 2.9

Table 6.5: ACT-R parameters for simulation of the metaphor-learning task.

6.3 Metaphor Comprehension in Context

In Section 6.1 we saw that, even though metaphor comprehension is generally as fast as literal comprehension is, some studies (Gibbs, 1990; Onishi & Murphy, 1993) found that subjects take less time to read literal sentences than to read sentences containing an anaphoric metaphor. Both Gibbs (1990) and Onishi and Murphy (1993) attribute this difference to the types of metaphors used in various studies, suggesting that anaphoric metaphors (such as those used in Gibbs, 1990; Onishi & Murphy, 1993) are harder than *A is B* metaphors, and that the latter were used by most researchers who found no difference between metaphors and literals.

However, this argument fails for one of the first studies that demonstrated the equivalence between metaphors and literals — Ortony et al. (1978). Ortony et al. used anaphoric metaphors, as Gibbs (1990) did; however, there was one conceptual difference among the two studies: Ortony et al. used the same sentences (e.g., *The hens clucked noisily*) for the metaphoric and literal conditions, varying only the preceding context (see also Section 6.1), whereas Gibbs used two different sentences for the literal and metaphoric conditions (e.g., *The creampuff did not show up for the match* or *The fighter did not show up for the match*). Thus, Ortony et al.'s (1978) targets always made literal sense, whereas Gibbs' (1990) did not.

6.3.1 Behavioral Data (Budiu and Anderson, 2000)

Budiu and Anderson (2000a) followed up on the hypothesis that metaphoric sentences that make literal sense are different than other sentences containing anaphoric metaphors. They ran an experiment in which subjects read passages followed by target sentences. The target sentences had a noun + verb + ending structure; their type was obtained by manipulating the metaphoricity of the noun and of the verb and could be one of the following:

1. metaphoric-noun–metaphoric-verb (abbreviated as metaphoric–metaphoric)

During history seminars, a massive young man always yawned and never paid any attention to the discussions. He was a very good linebacker who had been all-state in football. The seminar always came after his training sessions, so he was very tired.	Every year the Localville Women's Society for Animal Protection has a meeting. They bring in snacks, eat, and report about what was accomplished during the year. But this year, a major discussion topic was the new city regulations that allowed people to buy live animals from ethnic food stores.

	Targets	
The bear hibernated in class	*metaphoric–metaphoric*	The hens clucked noisily
The bear slept in class	*metaphoric–literal*	The hens talked noisily
The athlete hibernated in class	*literal–metaphoric*	The women clucked noisily
The athlete slept in class	*literal–literal*	The women talked noisily

	Probes:	
The man dozed during the class	*true*	The ladies discussed loudly
The man daydreamed in class	*false*	The ladies sang loudly

Table 6.6: Sample materials from Budiu and Anderson, 2000.

2. metaphoric-noun–literal-verb (abbreviated as metaphoric–literal)
3. literal-noun–metaphoric-verb (abbreviated as literal–metaphoric)
4. literal-noun–literal-verb (abbreviated as literal–literal)

At the end of each trial subjects had to verify the truth of a probe sentence, based on the preceding context (formed by the passage plus the target sentence). Table 6.6 shows two sample passages and corresponding targets and probes. Budiu and Anderson measured reading times[3] for individual noun, verb, and ending components and accuracy data from the truth verification task[4]. The overall sentence-reading times were computed as the sum of the component times.

Tables 6.7, 6.8 and 6.9 show the latency data obtained in this experiment. The noun-reading times were significantly longer for metaphoric nouns than for literal nouns (see

[3]More precisely, they measured the interval between the onset of the words on the screen and subject's key-press.

[4]However, I did not attempt to model the accuracies, so I do not discuss them here in detail.

Table 6.7), suggesting that subjects had some difficulty in understanding the metaphors. This difference was preserved in the verb-reading times: participants were reliably slower for verbs that followed metaphoric nouns, possibly due to a spill-over effect from the noun[5]. However, surprisingly, the endings of sentences with metaphoric nouns were read significantly faster than the endings of the literal-noun sentences (see Table 6.9). The short ending-reading times for metaphoric-noun targets countervailed the long noun- and verb-reading times; thus, there was no significant difference in the overall reading time for various target types (see Table 6.8). The results for the sentence reading times confirmed previous findings by Ortony et al. (1978) who showed that, when preceded by a long context, metaphoric targets are as quickly understood as literal targets. However, the metaphoric nouns influenced the accuracies (which were lower for metaphoric-noun sentences) and the component reading times.

Based on these data, Budiu and Anderson concluded that subjects had only a partial understanding of the metaphors and that, sometimes, they failed to integrate the metaphoric sentences with the preceding context (and thus read the endings faster). An analysis concerning the endings of target sentences confirmed this conclusion. As seen in Table 6.6, the endings of the target sentences could be split into two classes: one containing endings related to the passage (e.g., *class* for the targets of the linebacker story in Table 6.6) and another containing endings that were novel with respect to the passage (e.g., *noisily* for the targets in the women story from Table 6.6). When looking at reading times for the two classes of endings, Budiu and Anderson found that subjects were faster for the unrelated endings in the metaphoric-noun conditions (see Table 6.9). They argued that the unrelated endings offered little help in the process of integration with discourse; therefore, subjects may have failed to generate integrations for at least some of the metaphoric-noun sentences with unrelated endings and, thus, may have processed them quickly[6].

To summarize, Budiu and Anderson's (2000a) study suggests that, although people can read anaphoric metaphors as fast as literals, sometimes there is a comprehension cost that they pay in terms of accuracy and coherence with discourse.

6.3.2 Simulation for comprehension of metaphors in context

The basis for simulating comprehension of targets in Budiu and Anderson's (2000a) experiment is described in Section 5.2. We saw that the given information is used to find an interpretation in the context; even though that interpretation may match the novel sentence only partially, it can be used as a hook for integration with discourse. Thus, the total amount of given information in a target sentence is critical for relating it to the context. Sentences containing metaphors and unrelated endings have the least given information —

[5] It is possible that part of the processing load imposed by metaphoric noun was transfered to the following verb.

[6] However, there was no effect of ending relatedness on accuracy.

	Data	Model
Noun	Noun RT	Noun RT
Metaphoric	664	695
Literal	634	607

Table 6.7: Noun reading times (ms) from Budiu and Anderson (2000) and model results. ("RT" stands for "reading time.")

		Data		Model	
Noun	Verb	Verb RT	Sent RT	Verb RT	Sent RT
Metaphoric	Metaphoric	563	1919	556	1981
	Literal	559	1934	551	2054
Literal	Metaphoric	539	1940	527	1944
	Literal	526	1918	517	1930

Table 6.8: Verb and sentence reading times (ms) from Budiu and Anderson (2000) and corresponding model results.("RT" stands for "reading time.")

their given information may be determined only by the similarity between the vehicle and the topic of the metaphor(s); therefore, they have the least chance of being integrated with the preceding context. Note that the topic of the noun metaphor is present in the passage (e.g., *bear* and *linebacker* in the first column of Table 6.6), but the topic of the verb most often is not (although there is some word related to the topic — e.g., *hibernate* has the topic *sleep*, which is not present in the story in Table 6.6, although related words such as *tired* and *yawned* are). This asymmetry between the noun and the verb is carried through the literal targets as well (i.e., the literal *athlete* has a direct antecedent, but the verb *sleep* has not) and implies that, in the context of the experiment described by Budiu and Anderson (2000a), there is usually more given information in the noun than there is in the verb. In the following paragraphs, I use the term **antecedent of the verb** to mean the word in the passage that is most related to the literal topic (e.g., for the linebacker story in Table 6.6, the antecedents of both *hibernate* and *sleep* are either *yawn* or *be-tired*).

I discuss the predictions for the each component reading time separately, starting with those for ending-reading times (which are fundamental to this model). The predictions for the sentence-reading times are a direct consequence of the component predictions (as they result from summing up the components).

Ending-reading times. Figure 6.5 shows how the model comprehends metaphoric-noun–metaphoric-verb sentences. Parts a and b of Figure 6.5 present possible interpretation

	Data End RT			Model End RT		
Noun	Related	Unrelated	Average	Related	Unrelated	Average
Metaphoric	784	701	743	800	738	769
Literal	769	789	779	813	803	808

Table 6.9: Ending reading times (ms) from Budiu and Anderson (2000) and model results. ("RT" stands for "reading time.")

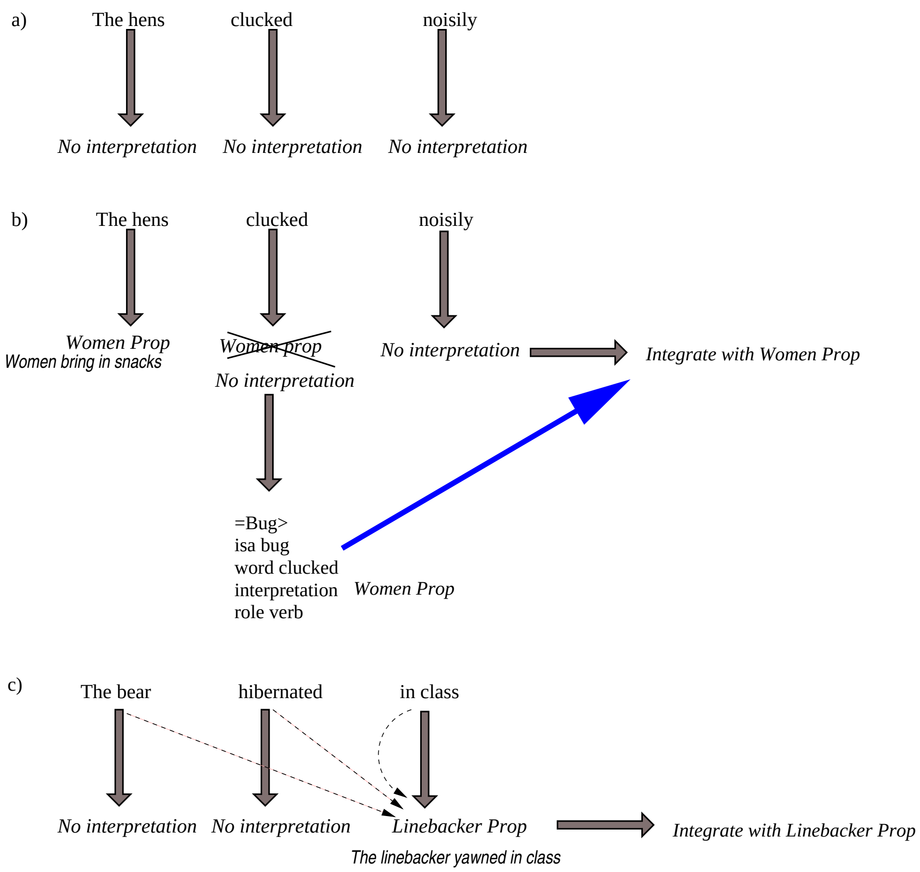


Figure 6.5: Comprehension of metaphoric–metaphoric sentences. a. Unrelated ending; no context integration. b. Unrelated ending; context integration. c. Related ending; context integration.

sequences for sentences with unrelated endings. For metaphoric–metaphoric sentences with unrelated endings, the most frequent case is when the given information does not suffice

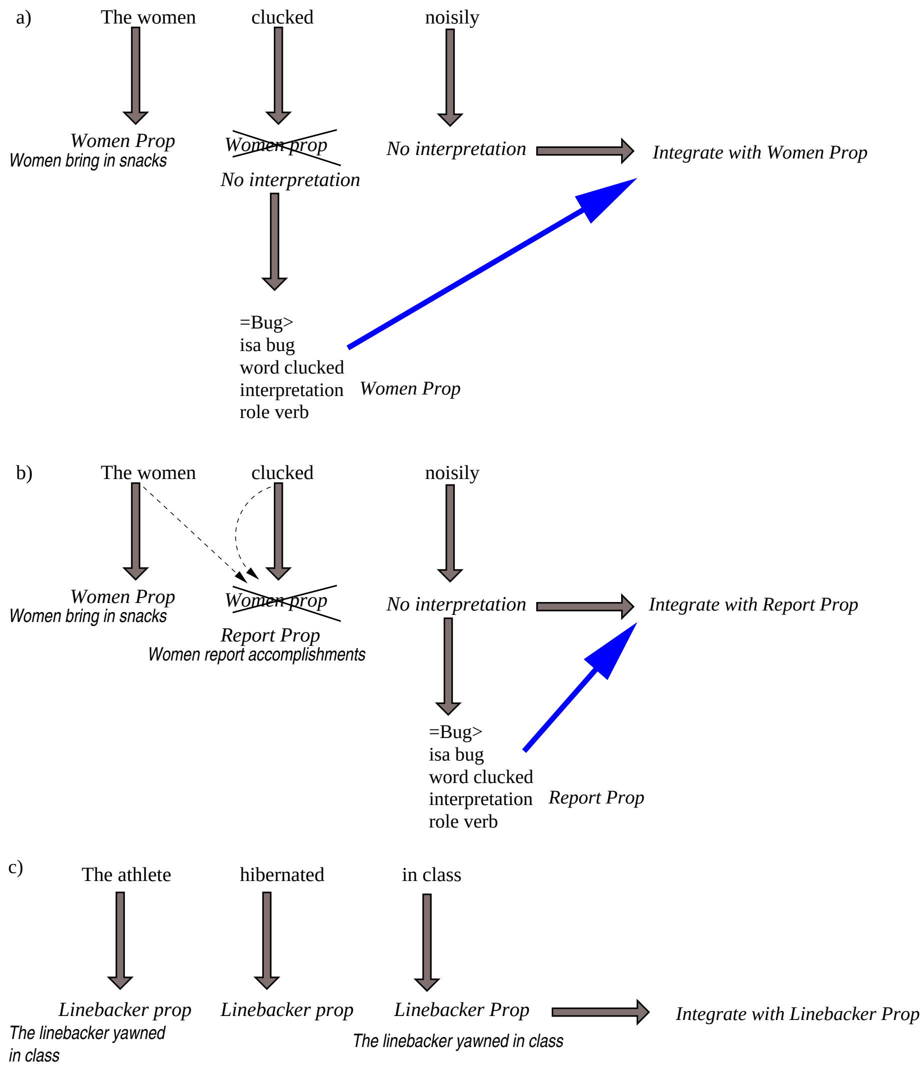


Figure 6.6: Comprehension of literal-noun targets. a. Unrelated ending. c. Related ending.

to retrieve any candidate interpretation from the context (see Figure 6.5a): the activation spreading from metaphoric words (e.g., *hens* and *clucked* in the figure) is not enough to select any proposition. Therefore, because it was not able to find a candidate interpretation at any point during the processing of the input, the model fails to relate the input to the context: the sentence is perceived as isolated. A less frequent case is when the similarity between the vehicles and the topics of the noun metaphor (e.g., between *hens* and *women*) is high enough as to make the model identify the referent of the metaphor when it is encountered[7] (i.e., to retrieve and validate a proposition about *women* when the word *hens* is read). In that case, the model may reject the proposition on the verb (unless it involves the antecedent verb and, even if it does, the model may reject it due to the low similarity between the related verb and the input verb — e.g., between *cluck* and *report*); however, the rejected candidate interpretation is recorded into the bug formed on the verb and thus can be retrieved at the end of the sentence to serve as a hook for integration with the context.

On the other hand, if, as in Figure 6.5(c), the ending is related to the context, it can contribute significantly to the finding of a hook for integration. Indeed, although no interpretation may be found on the metaphoric noun and metaphoric verb, the presence of an ending directly related to the context may boost the spreading activation to a level high enough to select a candidate interpretation from the discourse.

Whenever the model has a hook or a final interpretation, it uses it for integrating the input with the context. Context integration is quite minimalist: it means only updating the interpretation slots in the proposition links to point to the hook (see Section 5.2); in a more complex model of discourse processing it may involve more elaborate processes. However, the important assumption for simulating this experiment is that it takes extra time at the end of the sentence. Therefore, the model predicts that, each time when a hook can be found, context integration augments the ending-reading times. Hence, unrelated endings of metaphoric sentences should be read faster than related endings, because the latter lead more often to context integration than the former.

As discussed at the beginning of this section, the similarity between the literal verb and its antecedent in the story is small; for this reason, the literal verb does not help much with finding a candidate interpretation in the case of unrelated-ending metaphoric targets. For metaphoric targets with related endings, the chance of finding a candidate interpretation increases if the verb is literal. Thus, the same prediction stands for both metaphoric–metaphoric and metaphoric–literal targets: related endings promote context integration more than unrelated endings do and, therefore, the former take longer.

Whether the verb is metaphoric or literal, for literal-noun targets the model can always find a candidate interpretation, irrespective of the type of ending. Figure 6.6 presents a sequence of candidate interpretations for literal-noun sentences. Parts a and b of Figure 6.6 show two possible processings of a literal-noun target with an unrelated ending; in both cases, the literal noun (*women*) has an antecedent in the context; therefore, the model is able to find a candidate interpretation (any proposition involving *women*). Whatever happens on subsequent words, that candidate interpretation may serve as a hook and ensure integration. Thus, on the verb, the candidate interpretation may be rejected or accepted. If it is rejected, the model may either fail (see Figure 6.6a) or succeed (Figure 6.6b) to find

[7] Although the similarity is always fixed within a simulation, the activation noise can sometimes help the retrieval.

	Ending	
Noun	Related	Unrelated
Metaphoric	0.76	0.50
Literal	1.00	1.00

Table 6.10: Probabilities for context integration at the end of sentence, as predicted by the model.

another interpretation. In the latter case, the ending is a novel word that cannot match any proposition in the passage, so the candidate interpretation is rejected. However, the bug formed at the time of rejection (be it on the verb, as in Figure 6.6a, or on the ending, as in Figure 6.6b) contains a pointer to the candidate interpretation at the failure point (in the slot *interpretation*), so that candidate can be used for context integration.

Figure 6.6(c) shows the processing of a literal-noun target with a related ending. The only difference from the unrelated-ending cases depicted in Figure 6.6(a) and (b) is that, for related ending, the model has an interpretation at the end of the sentence. That final interpretation, rather than a previous candidate interpretation subsequently rejected, is used for integration. The case of literal-literal sentences is identical with that of novel sentences, discussed in Section 5.2, so I do not present it.

To summarize, the metaphoric-noun sentences with unrelated endings have the least given information and are the hardest to integrate with the preceding context. Therefore, the model often skips context integration for these sentences and, consequently, is faster to read their endings. In all other cases (metaphoric-noun targets with related endings or literal-noun targets), because more given information is present, at some point during the processing of the sentence, the model can find a candidate interpretation, which (whether or not subsequently rejected) can serve for context integration. The probabilities of integration in the different conditions are given in Table 6.10. The literality of the verb does not affect much the probability of finding a candidate interpretation, due to the low similarity between the verb and its antecedent. The predictions of the model for the ending-reading times are given in Table 6.9.

Noun-reading times. The data in Table 6.7 indicate that people take longer to read the initial metaphoric noun than to read a literal noun. The model does not naturally exhibit this behavior: in the case of metaphors, it fails to find any interpretation in the context. Because, after finding an interpretation, the model would need to match it and because in the case of literal nouns the model would always find an interpretation (see Figure 6.6), it would take longer to read a literal than to read a metaphor. To overcome this difficulty, I introduced an extra production, *Find-Antecedent*, for finding an interpretation; this production can fire only on the first word (Table 6.11) and competes with the other productions described in Table 3.3. The production *Find-Antecedent* is similar to the production *Successful-Match* in Table 3.4 and it combines searching for an interpretation with matching in one single step. Such a contraction is possible at the beginning of the sentence because the search cannot be helped by the spreading activation from the preceding words (there are no preceding words)[8]. The production *Find-Antecedent* has a probability of firing comparable with that of *Find-Interpretation* and higher than that of *Stop-Search*. Therefore, the *Find-Antecedent* production performs two functions: it speeds up the processing of literal nouns (by combining searching for an interpretation and matching into one single step), and it slows down the processing of metaphoric nouns (because this production must be tried and must fail, before the *Stop-Search* production successfully fires).

To summarize, even though the sentence processing model, as described in Chapters 3 and 5, does not naturally predict an advantage for the processing of literal nouns, by introducing an extra production, *Find-Antecedent*, which speeds up the processing of literals, the model is able to account for the pattern in the data (see Table 6.7 for actual numbers).

Verb-Reading Times. Again, the natural behavior of the model does not agree with the verb-reading data in Table 6.8, in which there is a significant effect of noun metaphoricity on verb-reading times. As discussed before, for literal nouns the model can find an antecedent and therefore a candidate interpretation; however, as parts a and b in Figure 6.6 show, that interpretation is very likely to be rejected, because the chance that is the "correct" interpretation (i.e., the one that matches also the verb) is small. The rejection and the search for another interpretation are time consuming. On the other hand, for metaphoric nouns the rejection cost is not paid, because, usually, no interpretation is found on the first word.

To compensate for these natural tendencies and in the spirit of the spill-over explanation discussed in Section 6.3.1, I introduced one production that attempts to find one antecedent for the noun while the verb is being processed, if no such antecedent was found before. The effect is that, after the verb was input, the model still ponders over the metaphoric noun, for which it has not found an antecedent. Such pondering results only in a delay, as the production *Noun-Spill-Over* (Table 6.12) is never actually fired[9], but is tried and its matching time is added to the time of the other productions.

Another modification was to increase the likelihood of not searching for an interpreta-

[8]The main reason for not doing this contraction in the production *Find-Interpretation* from Table 3.3 is that all previous words in focus can spread activation to a proposition to which they are related, but only one word can generally spread activation to a proposition link (unless all the words in the focus are related, which is an unlikely case)

[9]Note that production *Noun-Spill-Over* attempts to retrieve a sentence link containing the same word as the word that appeared in the input; such an endeavor is futile for metaphors.

```
production Find-Antecedent
 if  =goal>
       isa comprehend
       word =word
       role =role
       word-1 none
       word-2 none
       previous-interpretation none
       interpretation none
     =prop-link>
       isa prop-link
       parent =int
       - parent =goal
       type =role
 then
     =goal>
       interpretation =int
     =int>
       isa comprehend
       last-user [a] =goal
```

[a]In the actual model we use the *word* slot to keep the last user of an interpretation.

Table 6.11: Alternative production involved in the search for an interpretation on the first word.

tion, if another one was found before and rejected. The production *Prioritized-Stop-Search* (see Table 6.12) implements this modification and is very similar to the original *Stop-Search* production in Table 3.3; it is equivalent with increasing the granularity of the process of searching for an interpretation — instead of searching after each word, skip the search on the verb[10]. *Prioritized-Stop-Search* applies on the verb only if an interpretation was found on the preceding noun (which happens mostly for literal nouns); hence, the production *Prioritized-Stop-Search* speeds up the processing of verbs preceded by literal nouns only, by skipping the process of searching for another interpretation, after the previous one was

[10]This production may apply on the last word, too, but it cannot apply twice in a row — that is, on both the verb and on the last word. When it fires on the last word, the existence of a previous interpretation (which is a condition for *Prioritized-Stop-Search* to fire) ensures that context integration can be performed, although possibly suboptimally.

```
production Noun-Spill-Over[a]
 if =goal>
      isa comprehend
      word =word
      word-1 word-1
      - word-1 none
      word-2 none
      previous-interpretation none
      interpretation none
    =prop-link>
      isa prop-link
      - parent =goal
      parent =int
      child =word-1
 then
    =goal>
      interpretation =int
    =int>
      isa comprehend
      last-user =goal
```

```
production Prioritized-Stop-Search[b]
 if =goal>
      isa comprehend
      word =wd
      role =role
      task ``interpretation"
      interpretation none
      previous-interpretation =prev-int
      - previous-interpretation none

 then
    =goal>
      task "read"
      interpretation none
      previous-interpretation none
    =bug>
      isa bug
      word =wd
      role =role
      context =goal
      interpretation =prev-int
```

[a] In the actual implementation, there are more retrievals that regulate the amount of chunk-matching time spent in this production and thus the magnitude of the spill-over effect.

[b] See Footnote *a*.

Table 6.12: The productions involved in the processing of the verb.

rejected.

To conclude, the model, as described in Chapters 3 and 5, predicts that there would be a reading-time advantage of verbs preceded by metaphoric nouns over verbs preceded by literal nouns. To countervail this advantage, the model makes use of two supplementary productions: *Noun-Spill-Over* and *Prioritized-Stop-Search*. The first production delays the processing of verbs preceded by metaphoric nouns and the second speeds up the processing of verbs preceded by literal nouns (by skipping the search for another interpretation). The results produced by the model are shown in Table 6.8.

Tables 6.13 and 6.14 show the values of the similarities and of the ACT-R parameters that were used to obtain the numbers reported in Tables 6.7, 6.8 and 6.9.

Similarity	Value	Example
metaphoric-noun–antecedent	0.27	between *bear* and *linebacker*
metaphoric-verb–antecedent	0.15	between *hibernate* and *yawned*
literal-noun–antecedent	0.75	between *athlete* and *linebacker*
literal-verb–antecedent	0.45	between *sleep* and *yawn*

Table 6.13: Similarities for simulation of the metaphor-comparison task. The examples use the stories in Table 6.6.

Parameter	Abbreviation	Value	
word reading time (s)	R	0.26	
latency factor (s)	F	0.03	(see Latency Equation 2.4)
activation noise	ans	0.16	
retrieval threshold	rt	-1.04	
probability to stop searching	stop	0.12	0.38 in other models
association increment	pi_{ia}	80.00	only for propositions (see also Equation 3.1)

Table 6.14: ACT-R parameters for simulation of the metaphor-comparison task.

The model of sentence processing presented in this dissertation offers an interesting view of metaphor comprehension: whether metaphoric sentences are isolated or embedded in discourse, the success of comprehending them depends on the sentential context. We saw that the sentential context (i.e., the amount of given information in the sentence) can speed up the processing of a metaphor, if it precedes it (see Section 4.1); that it can ensure successful verification of anaphoric-metaphor sentences (see Section 6.2) or that it can help at relating the sentence to the preceding passage (see Section 6.3). This model also shows that, when the sentence context is not sufficient, metaphoric sentences are not understood correctly, even though parts of them may be processed faster. Thus, whereas Budiu and Anderson (2000a) failed to provide evidence for the initial hypothesis that metaphoric sentences as those used by Ortony et al. (1978) are read intrinsically faster than anaphoric-metaphor sentences that do not make literal sense (as those in Gibbs, 1990), one possible hidden variable that could explain the contradictory results obtained by the two studies is the amount of sentential context present in the target sentences in the two experiments — that is, the targets of Ortony et al. may contain less given information than those of Gibbs. However, more careful experimental examination of this hypothesis is necessary. Given that accurate integration with context may lack for metaphoric sentences, another implication of this model is that studies of metaphor comprehension should collect accuracy measures after each sentence, to guarantee correct comprehension of the metaphoric materials.

Chapter 7

Empirical Evaluation: Sentence Memory

Since Bartlett (1932) a lot of studies demonstrated that prior knowledge influences the recall of text. In a very famous experiment, Bransford and Johnson (1972) showed subjects a passage either preceded or followed by a topic (e.g., *washing clothes*), or with no topic information associated[1]. Bransford and Johnson found that participants had difficulty in comprehending or recalling the passage when they had no topic information or when the topic followed the passage. This result indicated that people actively use their background knowledge in comprehension and that, in the absence of a guiding script (i.e., in the absence of prototypical knowledge about the situation described in the text) they become confused and encode poorly the passage read.

Not only can script knowledge help recall, but sometimes it can interfere with it. Owens, Bower, and Black (1979) found that when subjects were shown a series of distinct episodes (e.g., in the kitchen, at the doctor's, at supermarket) linked together by a setting mentioned before the first episode (e.g., referring to a pregnant college student), the setting modulated the recall. Namely, not only the recall improved in the presence of a setting, but also the number of setting-specific intrusions made by subjects increased (for instance, subjects recalled that the nurse in the story ran a *pregnancy test* instead of *usual procedures*).

In this chapter I describe a related text-memory experiment (Bower et al., 1979) that assessed the influence of prior script knowledge on recall. I also show how the sentence-processing model can help explain the results obtained by Bower et al. and argue that the structures formed by the model during text comprehension shape the process of text recall.

[1]The passage was written such that the topic could not be inferred from its contents.

The Doctor

John was feeling bad today so he decided to go see the family doctor. He checked in with the doctor's receptionist and then looked through several medical magazines that were on the table by his chair. Finally the nurse came and asked him to take off his clothes. The doctor was very nice to him. He eventually prescribed some pills for John. Then John left the doctor's office and headed home.

The Dentist

Bill had a bad toothache. It seemed like forever before he finally arrived at the dentist's office. Bill looked around at the various dental posters on the wall. Finally the dental hygienist checked and x-rayed his teeth. He wondered what the dentist was doing. The dentist said that Bill had a lot of cavities. As soon as he'd made another appointment, he left the dentist's office.

The Chiropractor

Harry woke up with a bad pain in his back again. He decided to go see a chiropractor that very day. He had to wait a long time. Finally, the chiropractic assistant finished and left him, and the chiropractor himself came in. The chiropractor carefully examined Harry by feeling all the bones in his back. Eventually Harry left the chiropractor's office.

Table 7.1: Three related stories used by Bower et al. (1979)

7.1 Behavioral Data (Bower et al., 1979)

Bower et al. (1979) showed that, when exposed to two or more versions of the same script (e.g., visiting a health professional), subjects are likely to make more intrusions in memory tasks than when they study only one version. In Experiment 3 from Bower et al. (1979), participants studied 18 different stories for 10 minutes; each story was based on a script and for each script, there could be one, two or three stories. The related stories (i.e., those based on the same script) were disjoint in terms of script actions (except for the initial and final actions, which were shared by all of them). Twenty minutes after after the study period, subjects had to recall the stories, using their titles as cues. Table 7.1 presents three related stories, all based on the Visiting–A–Health-Professional Script. For each story, subjects had 1 minute to recall it, by writing down component sentences.

Table 7.2 shows the recall rates obtained for each condition. When they had seen only one version of the script, subjects were less likely to recall script-consistent propositions that had not been part of the story. However, in the two- or three-variants conditions, subjects reported more often propositions that had not appeared in the original story, but

Number of script versions[a]	Data		Model	
	Stated actions	Unstated actions	Stated actions	Unstated actions
1	0.38 (3.03)	0.07 (0.80)	0.36 (2.88)	0.07 (0.80)
2	0.28 (2.27)	0.11 (1.26)	0.30 (2.40)	0.12 (1.44)
3	0.32 (2.56)	0.10(1.16)	0.27 (2.16)	0.13 (1.56)

[a]Number of script versions refers to the number of different stories using the same script.

Table 7.2: Rate of recall per script version adapted from Bower et al. (1979) and results of simulations. Number of actions recalled is shown in parentheses.

which were consistent with the script. For instance, if they studied both the doctor and the dentist stories, subjects may have recalled that the doctor set up another appointment, although this sentence was not part of the doctor story. Note also that the subjects show a decrease in the number of stated actions that they report. This effect is possibly caused by the fixed amount of time (1 minute) allocated for the recall of one story — if the number of facts recalled in 1 minute remains about the same in all conditions, it is natural that fewer stated actions are reported when more intrusions are made.

Experiment 4 in Bower et al. (1979) replicates this result in a recognition-memory task. According to Bower et al. (1979), the explanation for this recall pattern lies in the usage of the script for recall. In the next section, I elaborate this explanation in the context of my sentence-processing model.

7.2 Simulation of sentence recall

The structure of the task in Bower et al. (1979) implies two modeling stages: comprehension and recall. The comprehension stage can be modeled using the sentence-comprehension model described in Chapter 3. In this section I discuss mainly the structures produced by the comprehension model and the recall model.

Comprehension. Comprehension proceeds as described in Chapter 3: the model searches for an interpretation in the background knowledge and matches candidate interpretations to the current word. The comprehension process itself hardly influences the recall; however, the structures built during comprehension are involved in recall. Therefore, let us examine once more the knowledge structures involved in comprehension.

When the model processes a new input word, it creates a propositional link that records

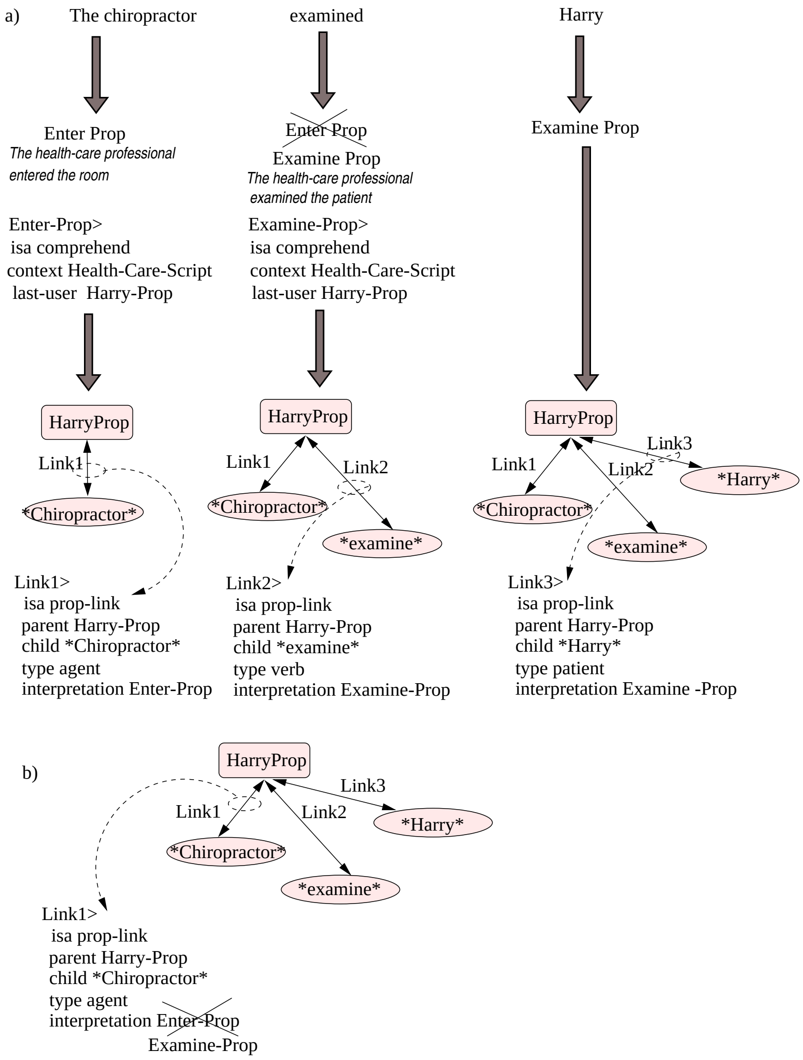


Figure 7.1: Products of comprehension. a. During the input-sentence processing. b. After integration.

the belonging of that word to the current sentence. The link contains a slot (*interpretation*) that must be filled with information about the interpretation of the sentence. At the time when the word is processed, the model takes care to update the slot *interpretation* to the current candidate interpretation. For instance, in Figure 7.1(a), when the model reads the first input word *chiropractor*, it retrieves the proposition *Enter Prop* as a candidate interpretation and sets the slot *interpretation* of *Link1* (the propositional link newly created for the current word) to point to this proposition. However, that candidate interpretation may be rejected further, if it does not match subsequent words. Thus, in Figure 7.1(a), the proposition *Enter Prop* is rejected and replaced with the proposition *Examine Prop*; this interpretation switch leads to an inconsistent representation for the input sentence — two links (*Link2* and *Link3*) point to the correct, final interpretation and another points to the first candidate interpretation, *Enter Prop*. Therefore, as we saw in Chapter 3, at the end of the sentence, during integration, the model spends time to ensure that all propositional links created for the input sentence point to the final interpretation (Figure 7.1b).

Note that in Figure 7.1(a), not only do the newly-created propositional links point to the current candidate interpretation, but that interpretation, too, keeps track of the sentence that used it as an interpretation (in the slot *last-user* — see also Table 3.3). The *last-user* pointer serves to avoid retrieving a proposition as a candidate interpretation again, after it was previously rejected. Thus, in Figure 7.1(a), although the proposition *Enter-Prop* is not the final interpretation, its *last-user* slot points in the end to the current sentence *Harry-Prop*, as does the same slot of the final interpretation *Examine-Prop*. This peculiarity of the model plays a role in recall.

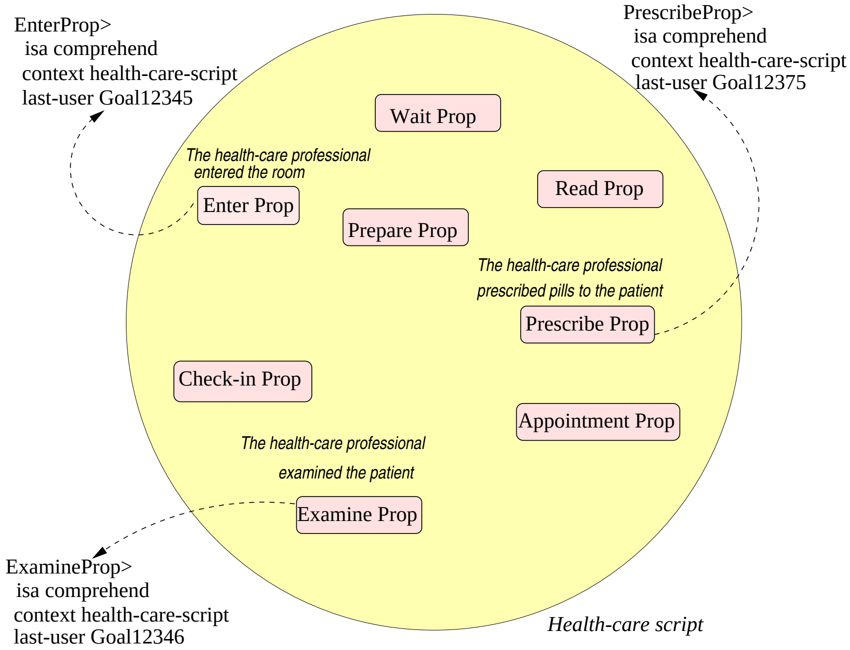


Figure 7.2: A script and propositions that are part of the script.

In this simulation, I assume that the background knowledge contains all propositions that are part of the pertinent script (e.g., all the propositions that are normally associated to visiting a health professional). These propositions are grouped together in scripts (Figure 7.2): in their chunk structure they keep a pointer to the script to which they belong (in slot *context* of the chunk comprehend). Given the design of Bower et al.'s experiment, it is realistic to assume that each sentence studied had a script correspondent; thus, in the comprehension phase, the model is able to find a script interpretation for each input sentence.

Recall. The recall part happens at an estimated average delay of 1743 s[2] after the study period of each story. For each story there is approximately 1 minute for recalling it. Subjects and the model were cued with the title of the story ("dentist" etc).

In this model, there are two types of recall: **cue-** and **script-based** recall. Intuitively, in cue-based recall, the model tries hard to remember a piece of information that was actually studied; in script-based recall, it reports a proposition that matches the story script and which was encountered at study (but not necessarily in the same story). When presented with a cue such as the story title, the model first attempts to retrieve a propositional link that was built at study and that involves the cue. If it cannot retrieve any propositional link containing the cue, it gives up. However, if it is able to find a proposition corresponding to a studied sentence, it uses it to report its components and to find out on what script the story was based[3]. Next, to recall a second sentence, the model can use cue-based recall (i.e., as before, it can attempt to retrieve another proposition based on the same cue), or, if cue-based recall is unsuccessful, it can use the script. Script-based recall consists of retrieving a script proposition that was used as (candidate) interpretation at study and was not recalled yet; specifically, it involves propositions whose slot *last-user* is initialized to some value. Script propositions are often rehearsed and, thus, they have high activations; therefore it is easier to retrieve script propositions (as in script-based recall) than to retrieve proposition links involved in the actual studied-sentence representation (as in cue-based

[2]This delay was computed by taking into account that at study and at recall stories are presented in random order. I estimated the time needed for studying one story as 66.67 s and, based on that and on the knowledge that the study phase lasted 20 minutes and that there was a 10 minutes interval between the end of the study phase and the beginning of the recall, I computed the average time interval between the the moment when one story was studied and the moment when the same story was recalled. Given that subjects recall a story for 1 minute, the average delay (in seconds) is given by the formula:

$$\begin{aligned} T &= \frac{1}{n^2} \sum_{\substack{i_1=1\ldots n \\ i_2=1\ldots n}} (1200 - 66.67(i_1 - 1)) + 600 + 60(i_2 - 1) \\ &= 1200 + 600 + \frac{n-1}{2}(60 - 66.67) \\ &\approx 1743 \end{aligned}$$

where $n = 18$ is the number of stories.

[3]Remember that, for each new proposition formed at study, the slot *interpretation* contains a corresponding script proposition. Once that script proposition is found, script identification is easy.

Parameter	Abbreviation	Value	
word reading time (s)	R	0.26	
latency factor (s)	F	0.04	see Latency Equation 2.4
activation noise	ans	0.40	
retrieval threshold	rt	-2.35	for comprehension
		-6.15	for recall
probability to stop searching	stop	0.38	
write time (s)	W	1.50	time to write a word
probability to retrieve by script	script	0.40	

Table 7.3: ACT-R parameters for simulation of Bower et al.'s (1979) experiment.

recall). On the other hand, although script-based recall is more successful than cue-based recall, it is also more imprecise and constitutes the main source of intrusions for this model. Indeed, if more than one story related to the same script is studied, then there are more script propositions that were used as interpretations and that can be retrieved at recall.

One peculiarity of this model is that it mistakes what the model thought during comprehension for what it was really shown (i.e., it may actually retrieve at recall script propositions that were not final interpretations at comprehension, but were considered as candidate interpretations — such as the proposition *Enter-Prop* in Figure 7.1a). This feature allows the model to make intrusions in the one-version condition; otherwise, in that condition there would be no reason to ever be mistaken, because all the script propositions that functioned as interpretations would have been correct.

Note that recall is conditioned on being able to retrieve an initial propositional link that belongs to a studied sentence, using the cue. If that link is not retrieved, the script cannot be accessed at all. To perform retrieval of the original-sentence links, which were created a long time ago at the beginning of the experiment and read about two times, the model needs to have a low retrieval threshold. On the other hand, at comprehension, the model needs a relatively high retrieval threshold, which is close to the base-level activation of the script chunks involved in finding an interpretation (because the retrieval threshold is used as a similarity threshold in the matching phase of the model: if a propositional link for an interpretation is under the threshold, then the interpretation will be considered invalid; otherwise it is valid — see Section 3.2 and Table 3.4). Thus, this tension between comprehension and recall leads to using two different values for the retrieval threshold in the two stages of the model.

The last two columns of Table 7.2 show the predictions obtained by the ACT-R model for the Bower et al.'s (1979) experiment. Table 7.3 presents the ACT-R parameters that were used for obtaining these numbers.

Chapter 8

Choices for Modeling Comprehension

Chapter 3 discussed the basic structure of the model and the representations on which it is based. In this chapter, I analyze which other modeling choices are available and their consequences on the model's performance. In particular, Section 8.1 examines possible representational alternatives and Section 8.2 looks at two other types of search processes.

8.1 Representation

In Section 3.1 we saw that the model uses an atomic representation for word meanings and a distributed representation for propositions. Alternatively, one could opt for distributed meanings and/or atomic propositions. Table 8.1 presents the matrix of choices for the propositional and meaning representations. Whereas the current variant of the model is in the atomic-meaning–distributed-proposition cell, past variants assumed a distributed-meaning–distributed-proposition representation. In this section, I analyze the implications that each representation choice has for the sentence-processing model and show that some of the cells are not practical.

Distributed Meaning Representation. The model described in this dissertation was originally based on a distributed meaning representation. When a distributed meaning representation is used, meaning is encoded as a set of semantic features (Figure 8.1). The features are not concentrated all in one chunk, but are connected to the meaning chunk via links. The meaning links are ACT-R chunks containing information about the nodes they connect and about the context in which they were last used. Retrieving the meaning of a word involves extracting an arbitrary number of meaning features. This representation

Proposition Representation	Meaning Representation	
	Atomic	Distributed
Atomic	has difficulties	N/A
Distributed	used	explored

Table 8.1: Possible representation combinations.

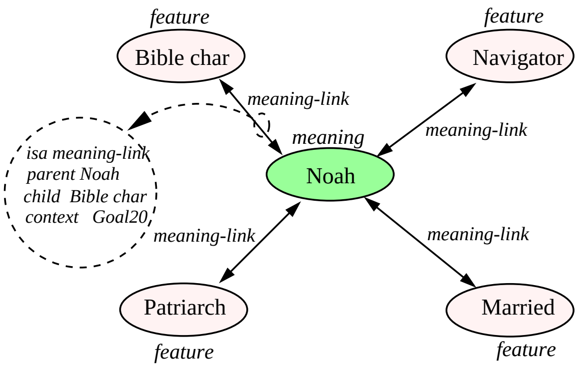


Figure 8.1: Distributed meaning representation.

offers a natural way of defining similarity between two words as the number of features that they share.

If the sentence-processing model were to use that representation, the basic search-and-match cycle described in Section 3.2 could still remain the same, but the unit of processing would become the meaning feature. Thus, instead of searching for an interpretation that involves the current word, the model could search for an interpretation involving only the current feature; each new feature extracted from the current word would be matched against the current interpretation[1].

One advantage of such a distributed meaning representation is that learning of new words becomes quite natural: each time a new word is encountered in context, one or several relevant features are picked up from the seed (i.e., referent) of the new word. With repeated usage, the right features may become the most salient; those features determine the associations with other words (i.e., the more features two words share, the more similar they are and the greater the strength of association between them, according to Equation 3.1). In Section 6.2.2, we saw that learning new meanings for metaphors is not plausibly modeled with an atomic meaning representation. In that model, associations between new meanings

[1]However, there are some potential changes in the process — for instance, one could choose to keep the last few features processed in the goal, separately from the previous words. These features would prime the selection of a new interpretation in the same way as the previous words do (but one can question whether so many items can be in the focus simultaneously).

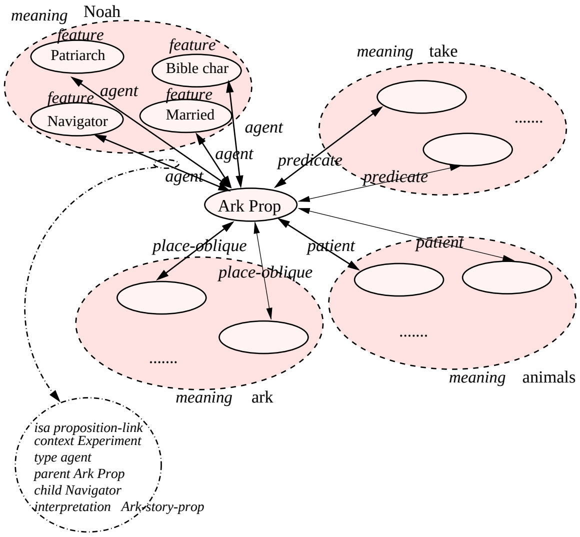


Figure 8.2: Distributed propositional representation with proposition links pointing to semantic features rather than meanings.

and old ones are instantaneously copied from the seed to the new meaning.

Another nice property of the distributed meaning representation is that it would allow a natural activation-spreading process. Note that, with the atomic meanings, similarities are computed based on some extrinsic semantic overlap; similarities between words and propositions build up on word similarities and activation spreading follows the similarity metric. If these similarities were based on shared features and if the propositional links (see Figure 3.1) were connected to features rather than meanings (as in Figure 8.2), then two propositions or meanings could spread activation to one other through common features[2].

Yet another advantage of distributed meaning representation is that it accounts nicely for the context-dependent variability in the word meaning. Certain contexts may increase the salience of one word feature — for instance, Anderson (1972) found that the sentence *Some pianos are heavy*, when used as a cue, can facilitate the recall of the sentence *The man lifted the piano*, but not the recall of *The man tuned the piano*. In Moses illusion tasks, the same distortion–undistorted-term pair does not work in different contexts[3]. Such aspects

[2]This chain-like spreading of activation would not be supported by the ACT-R architecture, though.

[3]For instance, people fall for the question *What is the title of the judge who heads the other six* (correct: *eight*) *on the Supreme Court?* about 18 percent of the time, whereas for the question *What is King Henry VIII of England famous for having eight* (correct: *six*) *of?* the illusion rate is about 86 percent (Lynne Reder, personal communication). However, with this particular example, an alternative explanation is that *eight* is more related to the the Henry-VIII context (because it overlaps with both *VIII* and *six*) than *six* is related to the Supreme-Court Context.

of language could be explained only rather circumventedly by an atomic-meaning model[4].

Unfortunately, a distributed meaning representation increases the processing time of the sentence-processing model considerably. Thus, if, on average, there are three features processed per word, the processing time of a model working at the feature level will increase roughly by a factor of three compared with the word-level model, because each feature will need to be retrieved separately[5]. This estimate is based only on the number of productions fired; a more complicate analysis, which depends on the details of the particular process used with this representation, needs to be done for the matching time[6].

To summarize, when compared with the atomic meaning representation, the distributed meaning representation is more appropriate for some tasks, such as word learning, but less feasible for fast sentence processing. For the purposes of the current sentence processing model, distributed representation at the symbolic level is too complex, and there is not enough time to retrieve it. It may be that the two representations coexist (with the atomic meanings capturing only a reduced, more salient number of semantic features) and the cognitive system uses one representation or another, depending on the tasks that it faces. Another possibility may be that the atomic representation is available at the symbolic level, but is actually based on a subsymbolic distributed representation[7]. Thus, similarity computation may reflect subsymbolic parallel processes that calculate the features shared by two objects.

Atomic Meaning Representation. This representation is described in Section 3.1, so I do not review it here.

Atomic Propositional Representation. An alternative way to representing propositions as graphs (see Figure 3.1) is to represent them as a single unit, by enclosing all relevant concepts within the same chunk. Table 8.2 shows one ACT-R chunk, *Ark-Prop*, standing for the proposition *Noah took the animals on the ark.* Note that in Table 8.2, meanings are atomic, too. Within ACT-R, atomic propositions are naturally paired with atomic meanings; thus, the distributed-meaning–atomic-proposition cell in Table 8.1 is not easy to implement, because it would imply a fixed number of semantic features (each of them corresponding to a chunk slot) allocated to each thematic role and would lead to

[4]One explanation could assume that variants of the same meaning actually occur in different propositions stored in long-term memory. The question how those variants were stored as different still remains open, though.

[5]Normally an ACT-R production cannot perform more than one retrieval.

[6]For instance, one could assume a smaller chance of selecting the right interpretation based on a single word feature than based on the entire word meaning.

[7]ACT-R has two levels of processing: the symbolic, production-system level, in which the computation is carried by productions and the subsymbolic, neural-like, parallel level. A lot of computations in ACT-R are performed by the subsymbolic level; for instance, the decision of which production to fire or of which chunk to retrieve are taken at that level.

```
Ark-Prop>
  isa proposition
  agent Noah
  predicate take
  patient animals
  place-oblique ark
  referent Ark-story-prop
  context Experiment
```

Table 8.2: Atomic representation for the proposition *Noah took the animals on the ark.*

huge proposition chunks. With an atomic representation, the proposition chunk must have one slot for each thematic role, which leads to large chunks with many slots often carrying no information. Moreover, with an atomic representation it is problematic to represent propositions in which multiple words have the same thematic roles (e.g., conjunctions or even noncoordinated words, such as *bus* and *Pittsburgh* in the sentence *In Pittsburgh you are not allowed to eat on the bus.*). Yet a third complication is that an atomic propositional representation predicts an all-or-none character to sentence recall: either all the words in the proposition are recalled or none of them is. This latter prediction is inconsistent with experimental results (e.g., Anderson, 1972).

The atomic proposition representation has the advantage of allowing a more traditional ACT-R solution to the problem of setting associations to reflect similarities. ACT-R normally regards associations between chunks as evolving from co-occurrences, rather than from similarities. As discussed in Chapter 3, to achieve the human-comprehension speed, my model uses the previous words in the focus to raise the activation of the right proposition and make it more likely to be selected as an interpretation. To allow for activation spreading from similar, but atypical meanings, to propositions, the ACT-R strengths of association, S_{ij} (see Chapter 2), were set to reflect similarities in the original model described in Section 3.2. Note that, by defining associations as similarities, in order to spread activation to the correct interpretation, words in focus need not occur frequently in the same context with the correct interpretation, but rather be semantically similar to that interpretation (e.g., for the concept *drops of molten silver* in *drops of molten silver filled the night sky* to prime the interpretation *Stars filled the night sky*, it is not necessary that *drops of molten silver* occurs often together with the proposition *Stars filled the night sky*, but rather that it is similar enough to that proposition).

However, if the propositions were atomic, then the same effect could be achieved by the ACT-R partial-matching mechanism (see Chapter 2), which allows the retrieval of a chunk that does not match exactly the conditions specified in the production, but is most similar to the chunk requested[8]. Thus, with partial matching turned on, a production like *Find-Interpretation* in Table 3.3 could require that the candidate interpretation to be retrieved involve the same words as the current input does; although a proposition satisfying all those constraints may not exist, the closest approximation would be retrieved. The partial-matching solution has other drawbacks, one being that the focus should keep also all the thematic roles of the previous words to know what to match against what (i.e., not any proposition involving any three words can be retrieved, but only one that involves those three words in some specified roles).

Distributed Propositional Representation. This is the representation used by my sentence-processing model and discussed in detail in Section 3.1. I must stress here that this representation has a very important implication for my model: it imposes an activation-spreading mechanism that is based on similarities, as suggested in the paragraph about atomic propositional representations.

In conclusion, each cell in the matrix of representation choices (see Table 8.1) represents a legal combination (maybe with the exception of the distributed-meaning–atomic-proposition cell). The atomic-meaning–atomic-proposition cell offers the advantage of partial matching as an alternative to associations as similarities, but is less plausible and puts too high a load on the goal structure. My model uses atomic meanings and distributed propositions, with the benefit of keeping the goal structure relatively simple and the associated comprehension process fast, but with the drawback of nonconventionally using associations to reflect similarities. Previously explored distributed-meaning–distributed-proposition implementations fare as more plausible on dimensions such as word learning, but are more complex and too slow. Perhaps the distributed- and atomic-meaning views could be reconciliated by assuming a subsymbolic distributed representation incarnated at the symbolic level in an atomic meaning representation.

8.2 Process

In Section 3.2 we saw that my model is basically a search-and-match process; in that section I also argued for the necessity of a search that is as informed as possible. In the model in this dissertation, activation spreading from other items in the focus ensures that the search

[8] Partial matching is not suitable for distributed propositional representations, because a partial-matching model would need that all the information inexactly matched be present in the same chunk that must be retrieved. Thus, to retrieve the proposition *The sky was filled with stars* based on the input *The sky was filled with drops of molten silver*, the model would require to retrieve a proposition chunk whose theme is *drops of molten silver*; to match that theme partially to *stars*, ACT-R needs to have the theme within the chunk that represents the proposition *The sky was filled with stars*.

process is quick and relatively informed. But one could decrease the reliance on associations, in favor of a more trial-and-error, procedurally-intensive process. Previous variants of this sentence-processing model explored different amounts of procedurality. In this section I review briefly two (extreme) alternative models (which were actually implemented) and their computational performance.

Purely Procedural Model. This procedural variant relies heavily on search and makes little use of cues such as the previous words. In this variant, there are no other words kept in focus except for the current word, so the search phase benefits only from activation spreading from that word. Moreover, the previous, rejected interpretation is not kept in focus either, preventing the model from benefiting from interpretation priming. Before accepting the most active interpretation as a candidate, the model must make sure that it matches previous words, which are extracted from memory. The search process in this procedural model is more precise than in the model described in this dissertation: because propositional links corresponding to previous words are extracted from memory, the model can also check whether those previous words appear in the same thematic roles in the interpretation and in the input sentence. Thus, one difference between the thesis model (as described in Section 3.2) and this procedural variant is that the latter is more robust to role confusions than the first. For instance, if the thesis model were to find an interpretation on the word *saw* in the sentence *The man the woman saw was tall* (and if it had no previous candidate interpretation in the goal), it would be equally likely to select *The man the woman saw was tall* and *The woman the man saw was tall*, because both interpretations would receive the same amount of spreading activation from the words in focus (i.e., from *man*, *woman* and *saw*). However, the procedural model would prefer the correct interpretation because it would check that all previous words have the same thematic roles in the sentence and in the interpretation.

Explicitly checking previous words without keeping them in focus has two drawbacks. First, it is time consuming, in the sense that there is a minimum latency corresponding to the chain of productions that fire to perform the check. The procedural model (as developed) attempts to limit this check by examining only a random number of previous words. Second, if there are many propositions that match the current word, the chance of retrieving the proposition that also matches the previous words is small; therefore, there will be a relatively large number of trials before finding the correct interpretation. The procedural implementation is worst in terms of scalability, because it only uses the current word as a cue for accepting an interpretation. Thus, if there are many propositions that match the second word of a sentence, the probability of extracting the interpretation that also matches the first word is small. For instance, if the first two words are *The water filled*, then when *filled* is read, there is no guarantee that the interpretation that is retrieved at this point also matches the previous word *water*, because, presumably, there are many propositions containing the concept *fill*. In fact, if there are m propositions matching *The water filled* and n propositions containing *fill* as a verb, the chances of retrieving a correct interpretation are m/n, with m potentially much smaller than n. The procedural model always detects whether the interpretation retrieved does not match the previous word, but then tries again to find a better interpretation. The second time the chance of retrieving the correct one slightly increases to $m/(n-1)$. For the simple case when $m = 1$, the expected number of retrievals until the correct interpretation is found is $(n-1)/2$. Thus, the more interpretations in the background knowledge matching the word *filled*, the higher the time for retrieving the correct one.

Purely Associative Model. At the other extreme, there is a model that eliminates completely any matching and relies entirely on associations from the words in focus. The backbone of such a model would be formed only by the production *Find-Interpretation*, as described in Table 3.3. Thus, instead of the three-step process described in Section 3.2, the associative model performs two steps:

Start with no candidate interpretation.

1. **Read.** Read next word.
2. **Search.** Search for a candidate interpretation; if none found either go to step 1 or to step 2.

As the model described in this dissertation does, this associative variant keeps the current word and two previous words in focus and, in the search step, retrieves the proposition with the highest activation as the candidate interpretation. However, the associative model completely ignores whether any of these words have the same thematic roles as those occurring in the input sentence. Also, because this model never matches anything, to avoid the situation in which one proposition is retrieved on the first word and kept up to the end, regardless of its correctness, the associative model needs to retrieve the most active proposition on each word. Because there is no matching involved, the associative model is somewhat faster than the thesis model (see Section 3.2) is[9], but it is also more vulnerable to role confusions.

To summarize, the comprehension model described in this dissertation could be stretched to become either entirely procedural or entirely associative. Relying more on trial-and-error search to find the right interpretation has the benefit of increased role precision, but is not fast enough to fit people's comprehension speed; moreover, the speed of a procedural model

[9] The associative model is not necessarily much faster than the original model is, because for each word the former needs to retrieve an interpretation, whereas, once it found the correct interpretation, the latter performs only matches.

depends on the contents of the long-term memory. At the other end of the continuum, an entirely associative model is fast, but exposed to grave role confusions. The dichotomy between procedural and associative variants is the dichotomy between serial and parallel search: the serial search implemented by the procedural variant consumes more time, but may be more exact; the parallel search of the associative variant is faster, but has less precision (due to being implemented entirely through ACT-R subsymbolic processes). It looks like a model of comprehension needs to place somewhere between these two extreme variants if it targets accuracy and speed comparable with those of humans.

Chapter 9

Computational Evaluation

Chapter 1 notes that, beside fitting behavioral data, a viable sentence-processing model must satisfy a number of computational criteria such as speed, correctness, and scalability. In this chapter I analyze the model in this dissertation in terms of these computational constraints.

9.1 Speed

People are very fast at comprehending language: normal reading happens at a rate of 500-700 ms per word. During this short interval a lot of processes must take place: perception and word encoding, syntactic analysis and semantic integration with the other words in the same sentence, reference resolution and binding if the sentence occurs within a text. The complexity of a realistic model of sentence comprehension is severely constrained by human's comprehension speed.

In Chapters 3 and 6 we saw that the model in this dissertation is able to produce overall sentence-reading latencies comparable to those of people. Thus, in tasks involving reading of isolated literal sentences (such as the Moses-illusion gist task, Reder & Kusbit, 1991 – see Section 4.2) the model reads an undistorted sentence in about 3.4 s (see Table 4.6), whereas in text-reading experiments, the model spends about 1.9 s for a literal sentence made of three content words (see Section 6.3 and Table 6.8). The latency difference from experiment to experiment is due to experiment specifics, but also to different sentence lengths and syntactic complexity (e.g., Reder & Kusbit, 1991 used sentences longer than three words and that were more complex syntactically than the targets in Budiu & Anderson, 2000a). Thus, whenever possible, one should look at the word-reading rate, rather than at the sentence-reading time. In my model, the word-reading latency has two components: first, an artificial reading intercept, reflecting the time to perceive and encode the word (and implemented as the effort parameter of the ACT-R production that inputs a new word into

Experiment	Reading intercept (s)
Gerrig and Healy (1983)	0.40
Reder and Kusbit (1991)	0.60
Budiu and Anderson (2001)	0.35
Budiu and Anderson (2000a)	0.26
Bower et al. (1979)	0.26

Table 9.1: Reading intercepts R in the simulations discussed in this dissertation.

the model) and, second, the time spent for finding an interpretation involving that word or for matching an existent interpretation to that word. The second component produces the latency variations among different experimental conditions (e.g., between metaphoric and literal sentences), whereas the first component is an experiment-dependent constant (word-reading time R in Tables 4.4, 4.9, 6.5, and 6.14) and is meant to encapsulate syntactic complexity, beside perception and encoding times. Indeed, remember from Chapter 3 that the model treats groups of words such as *drops of molten silver* or *animals of each kind* as a unique concept; therefore, for sentences involving such constructs, the word-reading intercept should be higher than for sentences such as *The women talked noisily.* In my simulations, the value of the intercept varies as shown in Table 9.1.

For the only experiment (among those simulated) that collected word-reading latencies (Budiu & Anderson, 2000a), the word-reading times predicted by the model matched closely those of human subjects. However, the caveat is that syntactic processing is not included in this model; therefore, a prediction of this model is that syntactic analysis happens at a rate smaller than the reading intercept R (see Table 9.1). Moreover, whatever additional processes (such as reference resolution or inferences) occur during text reading[1], their word-processing rates are also upper-bounded by R. Thus, given that syntactic processing and word perception and encoding are compulsory and given that the intercept R is rather small, another prediction of this model is that inferences during text reading are minimal, if any (because, otherwise, there would be too much processing burden for the intercept R to account for). In Section 7.2 (see also Anderson et al., in press for a more complete account) we saw that the assumption of minimal inferences during text reading is realistic, as the model was able to simulate sentence-memory results assuming no inferences at study.

To conclude, in evaluating a model, it is useful to compare its word-reading rates (or, if those are unavailable, its sentence-reading latencies) with those of people. However, a good match between the two does not automatically validate the model, unless the model completely accounts for all aspects of comprehension. In the case of the sentence-processing

[1]However, some of the experiments modeled do not involve such text processes (e.g., those using isolated sentences).

Experiment	Error rate
Gerrig and Healy (1983)	N/A[a]
Reder and Kusbit (1991)	0.030
Budiu and Anderson (2001)	0.037
Budiu and Anderson (2000a)	N/A[b]
Bower et al. (1979)	not available

[a]Only metaphoric sentences were considered; however, the average error rate was 0.002.

[b]All the sentence were novel.

Table 9.2: Error rates predicted by the model for literal, undistorted sentences having an interpretation in the long-term memory.

model described in this dissertation, the latencies and word-reading rates are similar to the behavioral data. Beside semantic-processing time, the word-processing time predicted by the model includes an intercept R, accounting for all other word processes that are not modeled. The value of this intercept reflects syntactic complexity of targets, but also upper bounds the amount of text-related inferences. Full validation of this model would mean proving that all processes that are not modeled by it can be accounted for at a rate of R per word.

9.2 Correctness

A model of sentence processing should be able to produce a correct sentence interpretation whenever humans do so. Therefore, the model should comprehend at least literal sentences with a simple syntactic structure. My model finds the correct interpretation in most such cases; Table 9.2 shows the error rates (i.e., rates of trials that end with no interpretation or with a wrong interpretation) for the literal, undistorted sentences for which an interpretation in the background knowledge exists.

Because it never checks whether a candidate interpretation contains the previous words (other than the current) in the same thematic roles as in the input sentence, the model may seem especially prone to role-confusion errors. These errors refer to the cases when the model finds an interpretation that contains the same words as the input, but with different roles — for instance, when the input *The man hit the ball in the park* is interpreted as *The ball hit the man in the park*. Previous-interpretation priming was one reason why such constructs did not create actual problems for the model. Thus, note that for sentences such as *The man hit the ball in the park*, the only point where an erroneous decision can be made is the word *park*. (For all the words before *park*, if the incorrect interpretation was chosen, it would be rejected because it would not survive the thematic-role match — e.g., *ball*, the patient of the input, would not match *man*, the patient of the incorrect interpretation *The ball hit the man in the park*.) However, if a candidate is rejected on the word *park*, it means that the previous candidate interpretation did not match that word, in spite of matching everything else[2]. Such a previous candidate might have been *The man hit the ball on the stadium*[3]. But that previous candidate is more similar to the correct interpretation *The man hit the ball in the park* than to the wrong interpretation *The ball hit the man in the park* (see Chapter 3). Hence the correct interpretation receives more spreading activation from the previous interpretation than the incorrect interpretation does and it is preferred.

Garden paths are instances of role confusions from which people (sometimes) recover. I do not discuss here the most famous example of garden path *The horse raced past the barn fell*, on the account that there is a fairly large number of native English speakers who do not understand that sentence easily. But there are instances of "unconscious" garden pathing — for instance, people experience a cost to understand sentences such as *The man hit by the ball was short*, due to expecting *the man* to be the agent of the verb *hit* (McRae, Spivey-Knowlton, & Tanenhaus, 1998)[4]. If the syntactic processor initially considers *the man* as the agent of the sentence and transmits this information to the model (which performs only semantic processing), then the model will find an interpretation in which *man* is agent. If, further on, after reading *by the ball* the syntactic processor decides that *ball* is the agent rather than *man*, the model will first check whether *ball* matches the agent of the current interpretation (which must be *man*) and then, due to matching failure, it will search for another interpretation in which *ball* is the agent. This interpretation switch may account for the delay experienced by people when reading such misleading sentences. However, there might be also more extensive repair concerning the roles that were first assigned incorrectly (e.g., update of the sentence link corresponding to *man* to reflect its new patient role). The current variant of the model does not accept error messages from the syntactic processor and does not check repeated assignment of a role, so it cannot perform any repair; however, an extension in that direction looks feasible.

9.3 Scalability

Cognitive models do not usually work with the same richness of knowledge as people do. Most models use modest long-term memories, containing only facts thought relevant for modeling one experiment. Although it would be a hard (and perhaps futile) endeavor to construct long-term memories that are comparable in richness to those of people, cognitive

[2] In this discussion, we assume that both *The man hit the ball in the park* and *The ball hit the man in the park* are part of the background knowledge.

[3] Note that any candidate interpretation abandoned at the word *park* must have matched *ball* as a patient.

[4] McRae et al. (1998) use the example *The cop arrested by the detective was guilty of taking bribe*, which can create problems for my model due to the similarity between *cop* and *detective*.

models must be fairly insensitive to the size of the knowledge base used: if one model's predictions strongly depended on having only a few sentences in the declarative memory, then that model would not be very valuable for predicting human behavior.

In all the simulations described in this dissertation, the knowledge-base used is on the order of a few propositions and concepts. In the previous sections we saw that the model is fast and correct on the most common sentences. In this section I examine the time for finding a sentence interpretation and the correctness of the final interpretation of a literal sentence on two large knowledge bases: the set of four-letter English words and a set of noun–verb–noun sentences. The first knowledge base is peculiar: each word in it is treated as if it were a sentence in long-term memory and each letter as if it were a word in a sentence. The position of the letter within a word corresponds to the thematic role. The second knowledge base is more conventional: it contains sentence fragments satisfying the pattern noun–verb–noun. These fragments were obtained by running a query on the Brown corpus (`http://www.ldc.upenn.edu/ldc/online/`) and were treated as separate propositions. I first ran my simulations on the four-letter–words database, thinking that a real propositional database may be hard to analyze (due partly to my model's dependence on a syntactical analysis and partly to it needing meaning similarities between words). In the end some of these difficulties were overcome and I was able to obtain more realistic estimations. However, I still present the word-database simulations because they offer interesting insights about the limitations of this model.

To avoid problems of computational tractability, I wrote a Lisp simulation of my model and fed it with the (word, respectively propositional) database[5]. The results reported in the next two sections are obtained by running the Lisp simulation for 500 times on an input selected arbitrarily from the database.

9.3.1 Knowledge Base of Four-Letter Words

In the simulations involving the word knowledge base, the long-term memory contains all four-letter English words[6]. Each word represents a proposition: letters in a word stand for concepts in the proposition and the letter position represents the thematic role. The simulation receives one word in the database as a input and finds an interpretation for that word. The process of finding an interpretation is that described in Chapter 3: at each moment, the last three letters read are in focus and they contribute (through spreading activation) to the activation of the words containing them; the most active word is selected as a candidate interpretation and then is validated if it matches the current letter. A previous interpretation that matched other letters is also kept in focus and spreads activation to

[5] Whereas ACT-R predicts the latency that humans would take for a given input, the time taken to make this prediction can be quite large for a big database. The Lisp simulation was constructed mainly with the purpose of avoiding this difficulty.

[6] Later we see that the database needed to be reduced to a subset of these words.

related words. Neither the Lisp simulation nor the model take into account the frequencies of words (or propositions), which may influence their base-level activations (see Chapter 2).

The word database is different in several ways from a four-concept–proposition database. First, words in propositions are more constrained than letters in words: for instance, certain words are only verbs or nouns or adjectives, whereas a letter can appear in any position within a word[7]. Second, the number of building bricks is very different: whereas there is a large number of different concepts that can be arranged in propositions, there are only 26 letters with which we can form words. Thus, one expects that the space of extant four-concept propositions be sparser than the space of possible four-letter words. An additional problem is that many letters occur more than once within a word; thus, if, say letter *l* is in focus, then words such as *lull* will be heavily primed and thus, on the average, preferred to all other words that contain only one *l*[8]. Given that most natural-language sentences seldom contain double words, I eliminated from the database all the words in which a letter appeared more than once; the unique-letter–words database had 967 entries. (The original number of words was 1179.)

In the rest of this section I describe the results obtained from running the Lisp simulation on the word database. First, I assume that the similarities between two letters are either zero (if the letters are different) or one (if the letters are the same); then, I discuss how the results modify if the similarities are continuous values between zero and one.

Extreme letter similarities (zero or one). I ran the Lisp simulation 500 times with random inputs for each set of parameters from various simulations described in this dissertation (Tables 4.4, 4.9, 6.5, 6.14 and 7.3). I collected several measures: the frequency of finding the correct interpretation, the average similarity between a wrong final interpretation and the correct one, and the number of interpretation switches per letter. This latter measure deserves some explanation: in Chapter 3 we saw that, if a newly selected interpretation does not match the current input, the model has the choice of either stopping the search and going to the next input letter (or word for sentence comprehension) or continuing the search for another interpretation. The probability of stopping the search is 0.38 throughout most simulations in the thesis (parameter *stop* in Tables 4.4, 4.9, 6.5, and 7.3), with the exception of the metaphor-comparison simulation (Table 6.14). An interpretation switch consists of rejecting the current candidate interpretation and replacing it with another one; interpretation switches are expensive and their number is the main variable that controls the latency predicted by the model for a given input. In Section 4.1, in the analysis of the latency predictions for the Gerrig and Healy's (1983) experiment, I assumed that, for each input sentence, there is at most one interpretation switch per word

[7]However, there are phonetic constraints that specify the combinations of sounds legal in a language.

[8]Even if three distinct letters were in focus, the words containing them may receive at most the same amount of spreading activation as *lull*.

Experiment	Accuracy	Switches	Similarity
Gerrig and Healy (1983)	0.88	2.10	0.64
Reder and Kusbit (1991), Ayers et al. (1996)			
literal	0.83	2.21	0.61
gist	0.84	2.25	0.62
Budiu and Anderson (2001)	0.83	2.30	0.62
Budiu and Anderson (2000a)	0.86	2.22	0.62
Bower et al. (1979)	0.84	2.29	0.62

Table 9.3: Results of the Lisp simulation on the word database, using parameters obtained from experiment simulations. *Accuracy* is the frequency of finding the correct interpretation, *Switches* is the average number of interpretation switches per letter, and *Similarity* stands for the average similarity between a wrong final interpretation and the correct interpretation.

(which is a realistic assumption for a small database). In the Lisp simulation for the word database, we see that, although, on average, there are more switches per letter, the number of switches is quite small even for a large database.

Table 9.3 presents the results of the Lisp simulations. For each experiment simulated in this thesis I estimated a set of ACT-R parameters; each row in the table corresponds to a Lisp simulation using that set of parameters. The accuracy column shows in how many cases (out of the total number of trials) the model found the correct interpretation for a word. The switches column records the average number of interpretation switches per letter. In those trials when the model did not find the right interpretation, I computed the similarity between the found interpretation and the correct one; the average of this similarity values is given in the last column of Table 9.3. The accuracy is above 80 percent in all cases and, on average, the model performs less than 2.3 interpretation switches per word; moreover, in those about 20 percent of cases in which it does not find a correct interpretation, the wrong final interpretation is about 60 percent similar with the right interpretation. Because two letters are similar only if they are identical and because the similarity between two words is computed by counting identical letters on the same positions (i.e., the similarity between *skin* and *spin* is 0.75, but the similarity between *skin* and *pins* is 0), an average similarity of 0.60 means that, on average, a wrong interpretation shares at least two letters with the correct one. These results show that the model can perform fairly well on a large database, both in terms of accuracy of comprehension and in terms of speed of comprehension. Also, to the extent that the word database is a pessimistic model of a propositional database (the latter being more constrained by word order and word morphology), the performance on a realistic propositional database may be even better.

To study how the size of the database affects the performance of the model, I eliminated

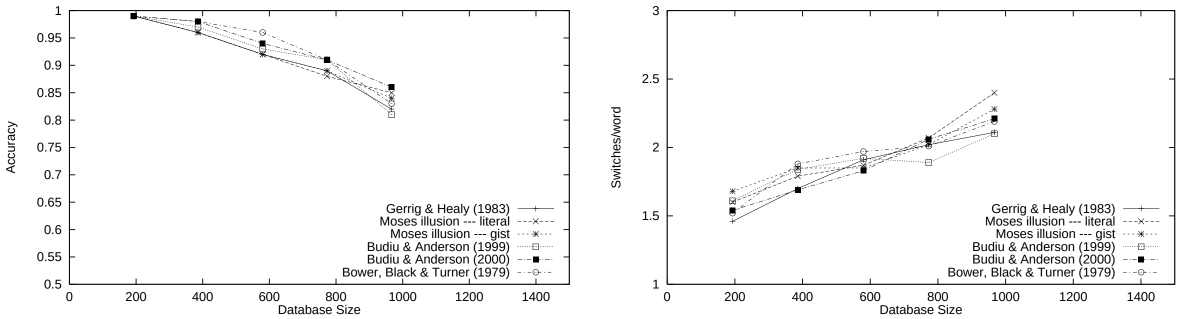


Figure 9.1: Performance of Lisp simulation on the word database as a function of database size. The results were obtained using extreme letter similarities. Different curves correspond to parameter sets from different simulations. a. Accuracy. b.Number of switches per word.

random words from the original set of 967 words. Figure 9.1 shows how the accuracy and the number of switches per word vary as a function of the database size. Ideally, the model's performance should be constant, independent on the size of the database. Unfortunately, that is not the case: the performance of the model tends to become better when the size of the database is smaller. From the sizes analyzed, we cannot say whether there is any asymptotic performance value to which the model converges.

However, these simulations assume zero similarity between different letters. When graphical characteristics of letters are used for defining letter-to-letter similarity, the results change substantially, as we see next.

Continuous letter similarities. In defining the letter-to-letter similarity, I used the same set of visual features as McClelland and Rumelhart (1981): namely, each letter was described by a subset of 19 visual features. The similarity between two letters is based on the ratio between the number of features shared by the two letters and the number of features possessed by either one of them (i.e., the ratio between the size of the feature-sets intersection and the feature-sets union). Because the average ratio was large (0.307), with about a third of the letter pairs having an intersection-to-union ratio higher than 0.35[9], the actual similarity between two letters was obtained by raising the ratio to the fourth power. This decreased the average letter-to-letter similarity to 0.04.

Table 9.4 shows the results obtained by the Lisp simulation that used this letter-similarity function. The accuracy in these simulation is between 57 and 72 percent (as opposed to the 80 percent in the extreme similarity setting from Table 9.3). However, the

[9] In the Moses-illusion simulation, the similarity between a good distortion and the undistorted term was estimated at 0.38; based on that, a similarity higher than 0.35 for so many letter pairs was considered too high and needed to be scaled down.

Experiment	Accuracy	Switches	Similarity
Gerrig and Healy (1983)	0.72	1.89	0.60
Reder and Kusbit (1991), Ayers et al. (1996)			
literal	0.58	2.06	0.59
gist	0.62	1.99	0.62
Budiu and Anderson (2001)	0.63	1.92	0.64
Budiu and Anderson (2000a)	0.69	1.81	0.63
Bower et al. (1979)	0.57	2.00	0.63

Table 9.4: Results of the Lisp simulation on the word database if letter-to-letter similarities are taken into consideration. *Accuracy* is the frequency of finding the correct interpretation, *Switches* is the average number of switches per letter, and *Similarity* stands for the average similarity between a wrong final interpretation and the correct interpretation.

average number of switches per letter remains low (circa 2) and the average similarity between a wrong final interpretation and the correct one is circa 0.60. The lower number of switches per letter (when compared to the corresponding values in Table 9.3) reflects the increased matching success. In investigating the reason behind the low accuracy, I discovered that there are eight letter pairs with similarity greater than 0.35; from the analysis of these pairs I noted that there are a few letters that are highly similar with several others: for instance, *c* is very similar with two other letters (*e* and *l*). If *c* were in focus, it would highly prime words containing more than one letter highly similar with itself: for instance, because the word *clue* contains the letters *l* and *e*, which are highly similar with *c*, *c* would prime *clue* more than *chin*, although they both contain *c*. This phenomenon is basically a repetition of the double-letter effect (in which one letter in focus primes mostly those words that contain it more than once). On the other hand, the reverse event can also happen: if two highly similar letters were in focus (e.g., *c* and *l*), words containing one of them would be doubly primed (i.e., *clog* and *chin* would be primed comparably, because, although the latter contains only *c*, it is primed by both *c* and *l*). Eliminating the words containing letter pairs with similarity higher than 0.35 increases the accuracy level with approximately 0.10. Moreover, the letter-similarity threshold is decreased from 0.35 to 0.20 (i.e., if we eliminate all the words that contain letter pairs with similarity higher than 0.20), the accuracy becomes higher than .70 for all parameter sets, but the database contains only 675 words.

The double-letter effect (in which a letter in focus primes words containing several letters highly similar with itself) is partly to be blamed on the similarity function between letters and words. In Section 3, I defined the similarity between a meaning and a proposition (i.e., between a letter and a word, in this chapter's terms) as being a weighted sum of the similarity between the meaning and each of the concepts composing the proposition. Thus,

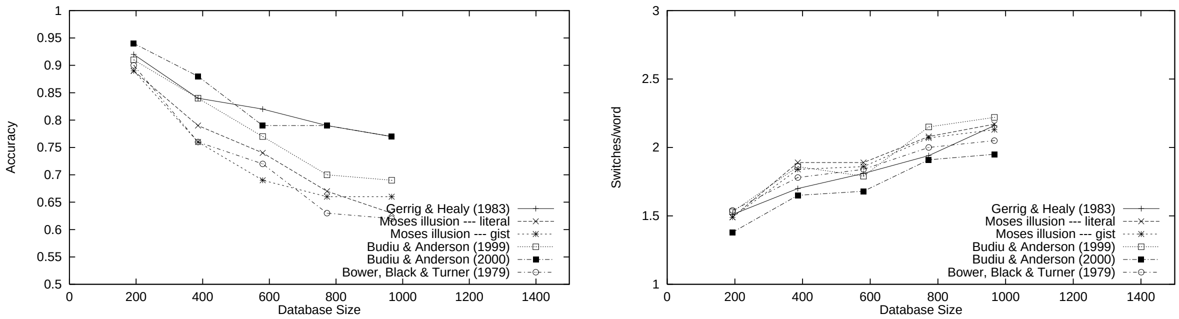


Figure 9.2: Performance of Lisp simulation on the word database as a function of database size. The results were obtained using continuous letter similarities. Different curves correspond to parameter sets from different simulations. a. Accuracy. b.Number of switches per word.

if letter c is in focus, its similarity to the word *clog* is $\frac{1}{4}(1.00 + 0.51 + 0.13 + 0.36) = 0.50$, where 1.00, 0.51, 0.13, and 0.36 are the similarities between c and c, l, o and g, respectively.

Figure 9.2 shows how the performance of the model is affected by the size of the database. The decrease in accuracy in Figure 9.2 seems to reach an asymptotic value at a database size of about 800 words, but databases larger than 1000 words are necessary to determine whether that trend in the accuracy curves is not local. Note also that different sets of parameters (corresponding to different simulations) have different scalability properties: in general, simulations with small values for the ACT-R retrieval threshold and the activation noise tend to be more scalable than the others, at least in terms of accuracy.

In conclusion, the Lisp simulation on the word database brings some useful insights: first, that the model can perform at decent levels of accuracy and speed if the similarities between letters are restricted to 0 and 1 values; second, that the performance of the model deteriorates if continuous letter-similarity values are used, because of priming of words that contain more than one letter similar to a letter in focus. This latter issue leads to the question of whether a typical natural-language proposition normally contains several items related to each other.

Another interesting lesson drawn from this simulation is that the retrieval threshold and the activation noise are two ACT-R parameters that affect most the accuracy of comprehension and that a large database may offer constraints for the values of these parameters. When I ran the word database first, I obtained quite poor accuracy for the parameters that I was then using for the sentence-memory and Moses-illusion simulations (even in the case of extreme letter-similarity values). It turned out that the performance could be improved by increasing the retrieval threshold and by decreasing the activation noise; interestingly, the modifications did not have a big impact on the actual predictions obtained by the

ACT-R simulations. The predictions reported throughout this thesis are those obtained after these modifications were made.

9.3.2 Sentence Knowledge Base

We saw that the performance of the model on the mock database of four-letter words was at the 60 percent level when letter similarities were values between 0 and 1. One reason for this behavior of the model was the double-letter effect: many words contained two or more similar letters and were favored by the presence of one of them in focus; also, similar letters in focus sent high activation to those words containing only one of them. In the word database, due to the existence of only 26 letters, the chance that one letter repeats within a word or that a word contains two similar letters is quite high; an interesting question is whether propositions expressed in natural language typically contain related concepts.

To answer this question, I collected a database containing 457 three-word sentences. This database was obtained using the web interface provided by the PennTreebank project (`http://www.cis.upenn.edu/~treebank/`). The PennTreebank project originates at University of Pennsylvania and annotates natural-language texts for linguistic structure. One of the corpora annotated by the PennTreebank project and which can be freely queried on-line is the Brown corpus. The Brown corpus was compiled in the early 1960s at Brown University under the direction of W. Nelson Francis and Henry Kucera. It contains 500 text samples of circa 2,000 words each, representing 14 categories[10] of American English texts printed in 1961. My model takes as input thematic roles for each word, whereas PennTreebank annotates sentences with syntactic labels (e.g., NP, VP) rather than thematic ones. Therefore I queried the Brown corpus for noun-verb-noun syntactic patterns, with the assumption that most results could be mapped onto the subject-predicate-object thematic pattern. I added some extra constraints to my search: that the nouns be non-proper, that the verb be in a present form (e.g., *write* or *writes*) or in a past-tense form (e.g., *wrote*), and that the last noun be preceded by the definite article *the*[11]. The result of the search consisted of 469 sentence-fragments; twelve of these contained words such as *other* or numerals used as words or words unlisted in the on-line version of Merriam Webster (`http://www.m-w.com`) and were eliminated. The remaining set of sentences consisted of 1023 words.

To run the Lisp simulation on this database I used inter-word similarities obtained based on Latent Semantic Analysis (Landauer & Dumais, 1997)[12]. LSA is a mathematical technique that generates a semantic space based on a text corpus. It represents meanings

[10] The categories include literature, fiction (with several subcategories), government, astronomy, American lore, press, religion, skills and hobbies.

[11] The online interface to PennTreebank accepts only search patterns that contain at least one specific word, beside syntactic nonterminals.

[12] Jose Quesada kindly collected the LSA similarities for me.

as vectors in a high-dimension space: each vector dimension is a different context and each vector value depends on the frequency of the word in the corresponding dimension context. Then LSA applies singular value decomposition to reduce the dimensionality of the semantic space. The similarity between two meanings is given by the cosine of the angle between the corresponding vectors in the smaller-dimension semantic space. Even though LSA starts with counts of occurrences of words in different contexts, the process of reducing the dimensionality of the semantic space successfully captures similarity relationships more complex than word co-occurrences. LSA was shown useful in simulating an impressive number of psycholinguistic phenomena such as vocabulary acquisition (Landauer & Dumais, 1997), emergence of natural categories (Laham, 1997), predication (Kintsch, 2001) and metaphor comprehension (Kintsch, 2000). It was also instrumental in rating the coherence of written text (Foltz, Kintsch, & Landauer, 1998), in selecting instructional texts appropriate to a student's level of knowledge (Wolfe et al., 1998), and in essay grading (Landauer, Laham, Redher, & Schreiner, 1997). For a relatively small set of words, LSA can be run on line at `http://lsa.colorado.edu`.

To obtain the similarity between any two words in the sentence database, LSA was run on the "General Reading Through First Year of College" corpus [13], resulting in a 300-dimensions semantic space. Twenty four words out of the original 1023 were eliminated because they did not occur in the corpus; the sentences containing these words were also removed from the sentence knowledge base (the final count of sentences being 436). Because an LSA similarity is a cosine (i.e., a number between -1 and 1), whereas my model works with positive subunitary similarities, the LSA similarities were transformed by the function:

$$s = \frac{1}{2}(1 + l) - 0.36$$

where l is the LSA similarity and s is the similarity used in my simulations. With this mapping, the maximum similarity between two words in the database was 0.61 and there were 3838 pairs of words with similarity over 0.35.

The results of the Lisp simulations on the sentence database are shown in Table 9.5. The correctness is always over 94 percent: the model is more successful on the sentence database than on the word database. Moreover, the number of switches per word is always less than one, as the model finds the correct interpretation on the second word in at least 79 percent of the input sentences. One could perhaps question the extent to which a database of 436 propositions is a good sample for the human memory. On the positive side, the sentence database included propositions with highly similar words (53 propositions contained at least two words with similarity higher than 0.35) and the average similarity between two propositions was 0.19; one word repeated at most nine times. These proportions may be

[13] The "General Reading Through First Year of College" corpus uses texts, novels, newspaper articles, and other information, from the TASA (Touchstone Applied Science Associates, Inc.) corpus used to develop The Educator's Word Frequency Guide.

Experiment		Accuracy	Switches	Similarity
Gerrig and Healy (1983)		1.00	0.46	N/A
Reder and Kusbit (1991), Ayers et al. (1996)				
	literal	0.97	0.72	0.47
	gist	0.94	0.68	0.36
Budiu and Anderson (2001)		1.00	0.69	N/A
Budiu and Anderson (2000a)		0.99	0.45	0.51
Bower et al. (1979)		0.95	0.70	0.44

Table 9.5: Results of the Lisp simulation on the three-concept–sentences database. *Accuracy* is the frequency of finding the correct interpretation, *Switches* is the average number of switches per letter, and *Similarity* stands for the average similarity between a wrong final interpretation and the correct interpretation.

more realistic than those corresponding to the word database. On the negative side, there was no consideration of word frequencies when building the database (although words with high frequencies tended to occur more often) and most words occurred only once.

To conclude the discussion on scalability, the model presented in this thesis has a very good performance (in terms of accuracy and latency) on a database made of hundreds of propositions. However, its performance is not as good if the knowledge base is made of four-letter words treated as propositions. Thus, statistical properties of natural language, such as number of propositions in long-term memory in which one word or a pair of words appear, the similarity between words within the same proposition, or the number of words in the vocabulary, play an important role in the success of this model. Also, the word-database simulations indicate that the model may have difficulties with propositions that contain the same word more than once.

Chapter 10

Comparison with Other Models

This chapter examines the ideas and mechanisms shared by my model with other models of text comprehension or of various behavioral phenomena presented in this dissertation. I discuss in relative detail Kintsch's (1988) construction–integration theory and also Myers and O'Brien (1998) memory-based processing model, derived from Kintsch's model. I also review theoretical explanations offered for metaphor-position effects and for Moses illusion, as well as the given-new theory.

10.1 The Construction–Integration Model

The construction–integration (CI) model (Kintsch, 1988) is a well-known theory of text processing, which has been applied to a number of domains. This theory consists of two phases: the creation of a text-base (construction) and then the selection of the elements relevant for comprehension (integration).

The CI theory represents knowledge as an associative network in which concepts or propositions are nodes; connections among nodes have positive or negative strengths. Both text base and background knowledge are encoded as knowledge nets; however, the text base is a structure separate from the background-knowledge net, although it is obtained by selecting and modifying propositions from the background-knowledge net. With this representation, the meaning of a concept is defined by the nodes activated in the net at the moment when that meaning is processed.

The construction phase uses sloppy production rules to build a rich and possibly incoherent text base; thus, several interpretations (some wrong) can coexist in the text base. For instance, for the sentence *The linguists knew the solution of the problem would not be easy*, the CI model first assumes that *the solution* is the object of *know* and then it forms another proposition in which *the solution* is a subject; however, at no point during construction is the old *solution*-patient proposition deleted from the text base. The propositions and concepts in the text base serve as independent retrieval cues for other propositions and concepts related to them. Beside these unstructured inferences, some more directed inferences may be generated (e.g., bridging inferences). In the final step of construction, the text base is converted to an association net: the nodes in the text base become associated based on strengths; two propositions from the text are associated according to their proximity in the text; background-knowledge propositions inherit their strengths from the background-knowledge base.

The integration phase consists of a connectionist-flavor process that eliminates the contradictory or incoherent elements from the text base. In the association network, activation spreads around until the system stabilizes; the activation spreading process is modeled by repeatedly multiplying an activation vector with the network connectivity matrix. If a fixed point is not reached, the construction phase is invoked again and new propositions are added to the net; then, the integration phase is started once more. At the end of the integration phase, the highly active nodes form the final text representation.

Before comparing Kintsch's model with the model in this dissertation, we must note, as Kintsch himself does, that the CI theory is "not concerned with specific strategies or rules for the construction of text propositions or inferencing", being intended more as a framework in which such strategies can be easily developed. Because the CI theory does not make any latency commitments and because it does not address the same behavioral datasets as this model, it is hard to compare them in terms of the evaluation criteria established in this dissertation. However, we can compare the mechanisms that the two models use.

First, both models are based on associations between words and propositions; nonetheless, the associations have a somewhat different semantics in the two models. Thus, in my model, they reflect semantic similarity between words; in the CI model they sometimes reflect text proximity of propositions.

In terms of processes, Kintsch's model works at a higher granularity than my model — it performs integration only at the end of the sentence (or at the end of an independent sentence fragment) and it does not commit to a final interpretation before the integration. Unlike my model, which selects one proposition as an interpretation and invalidates it as soon as possible, Kintsch's text base maintains a set of possible candidates for an interpretation regardless of their validity. The correct interpretation in my model is determined based on matching the current word against the current candidate interpretation and the success of match is determined by the success of retrieval. In the CI model, the final interpretation is found through the convergence of activation values of the elements in the text base; thus the composition of the text base (in other words, the context) can influence the meaning of a word (in the sense that those associates of the word that are related to the other elements of the text base end up with higher activation after the integration phase). My model is also sensitive to context, but in a different way: if the context is rich enough to provide the right interpretation for the sentence, a new word (be it literal or metaphoric) will quickly and correctly be added to that (see Section 4.1); if there is no context, or if the context is scarce, then an interpretation irrelevant for the actual sentence may be found. Thus, because it assumes atomic word meanings rather than defining meanings as sets of associates, my model would not predict that word meaning is dependent on the context[1], but rather that the proper context can speed up the processing of a word.

More recently, Kintsch addressed the problem of metaphor comprehension in the framework of his CI theory. Specifically, he looked at the comprehension of A–*is*–B metaphors. Kintsch used an LSA-based (Landauer & Dumais, 1997) knowledge representation, in which the strengths of the association-network connections are given by their LSA distance (see Section 9.3.2 for a description of LSA).

To apply the CI theory to this knowledge structure, Kintsch (2000) defines a predication algorithm for understanding A–*is*–B metaphors. According to this algorithm, the closest m ($500 < m < 1500$) neighbors of the predicate B are selected to be part of an association network, together with A and B. In this network, connections' strengths reflect the LSA distance between the corresponding words; in addition each node is connected by an inhibitory link to all the other nodes, so that the sum of all strengths in the network is zero. Integration is run on this network, in which the activation values of both A and B are kept constant to 1, to the effect that, when the system stabilizes, only those nodes related to both A and B have a positive activation value. The meaning of the metaphor is given by the centroid of A, B, and the most active k terms in the network. Kintsch argues that the same theory can be applied for understanding of literal predications.

In comparing Kintsch's metaphor-comprehension theory with the model in this dissertation, we must keep in mind that there are important scope differences between the two. Specifically, Kintsch's theory only addresses predicative metaphors, whereas my model addresses metaphoric sentences that can be predicative or anaphoric. Although this dissertation does not specifically discuss the case of predicative metaphors, they can be regarded as a particular case of metaphor-last sentences in the Gerrig and Healy (1983) simulation. Thus, for a sentence such as *Some jobs are jails* my model may end up with the interpretation *Some jobs are freedom-restraining*, based on the similarity between *jail* and *freedom-restraining*.

Kintsch evaluates his theory in terms of three empirical results that it captures: (1) metaphors are irreversible, (2) activating the literal meaning of the metaphor can harm the comprehension of the metaphor, (3) understanding metaphors is similar to understanding lexically-ambiguous words. Next I discuss how my model fares on the same tests.

The irreversibility of metaphors refers to the difference between *A is B* and *B is A* — compare, for instance, *Her surgeon is a butcher* with *Her butcher is a surgeon*, or *His*

[1] It is true, on the other hand, that those associates of the meaning that are related to the context are more active in my model, too, because they benefit from spreading activation from both the context and the meaning.

marriage is an icebox with *His icebox is a marriage.* Due to the asymmetry of Kintsch's predication algorithm, his theory agrees with the data. With respect to my model, at the end of the sentence *Her surgeon is a butcher*, if it has not found a valid interpretation yet, the model will look for a sentence in which *butcher* is part of the predicate. It could retrieve *Her butcher is a surgeon*, but that sentence would be rejected because *butcher* may fail the matching test. An interpretation such as *Her surgeon is rough* may be preferred, if *rough* and *butcher* are more similar than *butcher* and *surgeon.*

With regard to the literal-meaning interference with metaphor understanding, Kintsch suggests that preceding a metaphoric sentence such as *My lawyer is a shark* with a literal sentence such as *Sharks can swim* leads to people taking longer to understand the metaphor. In Kintsch's predication model, this effect can be simulated by asserting that *Sharks can swim* activates those neighbors of the predicate that are related to the literal meaning of shark; they start with some positive activation at the beginning of the integration and it takes longer for these priors to be washed out. In the current variant of my model, prior sentences do not affect comprehension, neither can activation of a chunk increase unless it is retrieved or related to some element into focus; however, if one assumed that the interpretation corresponding to the previous clause (*Sharks can swim*) was still in focus, then that interpretation could interfere with the retrieval of a correct interpretation for *My lawyer is a shark.* For the comprehension to succeed, the previous clause interpretation would need to be removed from the focus.

Another behavioral-data point cited by Kintsch as predicted by his theory is that metaphor understanding is similar to lexical-ambiguity resolution. Both Kintsch's model and mine predict that in the presence of a supporting context, the right meaning of a homonym is retrieved[2]. In the same spirit, Kintsch shows that his model predicts results by Gernsbacher and Keysar (1995); these authors have shown that subjects verify literal statements such as *sharks are good swimmers* faster when they are preceded by a literal statement (e.g., *The hammerhead is a shark*) than when they are preceded by a metaphoric statement (e.g., *My lawyer is a shark*). Again, my model can explain these data if the proposition corresponding to the previous sentence is kept in focus. In that case, for a literal sentence, the proposition in focus would be related to the statement to verify and would spread positive activation to the latter. On the other hand, for a metaphoric sentence, the interpretation corresponding to the metaphoric sentence (e.g., *My lawyer is ravenous*) is little related to the statement to be verified and thus would spread negative activation to that statement.

Beside the three types of experimental data that it captures, another feature of Kintsch's

[2]For my model, supporting context may prime the right meaning if one assumes that meaning extraction is influenced by the items in focus: thus, a candidate interpretation (and other words, too) in focus can spread activation to the meaning that is relevant to the context and increase its likelihood of retrieval. To account for lexical ambiguity, I must also define the similarity between a word link and a proposition or a meaning (which can be defined as for proposition links).

model is that it proposes the same comprehension process for both literal and metaphoric predication. This is also true of my model — if the metaphor occurs at the end of the sentence (and it does for predicative metaphors) and if the model has found the correct interpretation, it will integrate the metaphor smoothly into it, in the same way it would do with a literal sentence. If the correct interpretation was not found (i.e., the context was scarce), then the search for an interpretation occurs for both the metaphor and the literal. The only potential for a significant difference between metaphors and literals is when the similarity of the metaphor to the term it denotes is very small (see Section 4.1 for a discussion of acceptable similarity values).

In summary, my model passes the evaluation tests designed by Kintsch for his metaphor model, even though it was not conceived with these datasets in mind. Also, it is more general than Kintsch (2000)'s model, because it is not limited to modeling predication, which it treats as just one particular type of sentence. Kintsch's model of predication is different than CI in general, in that the association net is constructed in a peculiar way, not typical for normal sentences and in the integration phase the activation values of the A and B terms involved in predication are clamped to 1.

10.2 Memory-Based Text Processing

Although my model can understand sentences embedded in context, it is not a complete model of text processing. Thus, it does not deal with binding or pronoun resolution. However, it is interesting to compare it with existent text-processing theories.

One current view, popular among several researchers (Myers & O'Brien, 1998; Noordman & Vonk, 1998; Cook et al., 1998; Albrecht & Myers, 1998; Sanford & Garrod, 1998) is the memory-based text processing (MBP). The central assumption of MBP is that the processes involved during text comprehension are an effect of more basic memory processes: reading new information evokes older information (from the text) through an activation-spreading process and, thus, makes that information readily available. (This process is called **resonance**.) Whatever inferences are made during reading, they are not explicit, but rather due to the old information being "dumbly" activated. In this section, we discuss in some detail one instance of MBP modeling that belongs to Myers and O'Brien (1998).

In their model based on the CI theory, Myers and O'Brien (1998) use a knowledge network whose nodes are concepts, propositions and sentence markers (which represent local context markers, reuniting together a set of propositions that occur in the same sentence). The network contains edges between sentence markers and component propositions, between propositions and component concepts, and between pairs of propositions that are either identical or one part of the other.

The model assumes that working memory contains the sentence marker of the current sentence, "concepts and propositions derived from the sentence being read and similar elements carried over in working memory from previous sentences." The resonance process is a cyclic process consisting of activation spreading in the net. The activation starts propagating from the working memory; it is divided equally among all edges fanning out of a node and the activation spreading from a node is a fraction of the incoming activation to that node[3]. If the activation to be spread from a node decreases to a value below a certain threshold, that node sends out no activation at all. The resonance ends when no nodes can spread any more activation. Note that, although the incoming activation increases the total activation of a node, the total activation is not spread out, but only a fraction of what has been received at the previous cycle[4]. When resonance ceases, the active elements are those nodes with the highest total activation; they form the working memory together with concepts, propositions, and sentence markers from the next sentence.

Myers and O'Brien test their model against two empirical datasets: Albrecht and Myers (1995) and O'Brien, Plewes, and Albrecht (1990). Albrecht and Myers (1995) showed that a previous unsatisfied goal of a character in a story was not reactivated by an action that contradicted it, unless the text mentioned an object associated with the goal. For instance, subjects in Albrecht and Myers's (1995) experiment read a story about a captain who sat at his desk to make an inventory at the end of a cruise and who was interrupted during this activity. The story then mentioned that the captain came back in his office, sat at his desk and was happy to be done with the cruise. Thus, in this example, the last proposition contradicted the initial inventory goal; subjects were slower to read it when the previous sentence contained a reference to the captain's desk than when it did not. This slowdown was interpreted as reflecting the reactivation of the unsatisfied goal. Myers and O'Brien's model captured these results: before the reinstantiation sentence (mentioning the object associated to the goal), the goal and the associated concepts were not present in working memory, but they entered it after that sentence. However, if the sentence did not mention the object, the goal was not in the working memory.

The other experiment Myers and O'Brien simulated was that of O'Brien et al. (1990), who found that anaphoric references to more elaborated concepts (i.e., that occurred in more than one text proposition) were read faster than references to unelaborated antecedents. For instance, their subjects read a story about a a young girl who loved to play in the barn. Then the story elaborated on what the character did in the barn during several sentences. Further on, the text mentioned that the girl worked in the church, with no subsequent elaboration. Later on, subjects read more slowly a sentence involving the church than one about the barn. Myers and O'Brien's model kept the antecedents longer in the working memory, even after sentences unrelated to them, thus suggesting that the faster reading times of the references to them were due to multiple ways of access created through elaboration.

[3]The size of the fraction exponentially decreases with the "depth" of the node in the activation chain.

[4]A cycle consists of activation spreading from one node to all nodes directly connected to it.

It is difficult to compare directly my model with Myers and O'Brien's model, because my model makes no use of concepts such as working memory and the high activation of an item is transparent to the model unless it retrieves that item. The two models are similar in the sense that all the inferences performed (as well as the interpretation found for the sentence in the case of my model) are implicit and derived from the contents of the focus or, respectively, of the working memory. However, the working memory in Myers and O'Brien (1998) is much larger than the focus in my model and the process of activation spreading is limited to one cycle in my model.

The following discussion attempts to show how my model could account for the data addressed by Myers and O'Brien and contains a lot of speculation, as the exact details depend on exact activation values. With respect to the Albrecht and Myers's (1995) data, let us make the same assumption that we made in the comparison with Kintsch's (2000) model — namely, that the model keeps in focus the interpretation corresponding to the previous sentence. If a novel sentence mentions an object uniquely associated to the unsatisfied goal, the model may end up using the goal proposition as a hook for that sentence[5]. If the goal proposition becomes a hook and if it is kept in focus, the processing of the contradictory sentence may be delayed to the extent that the goal in focus negatively influences the comprehension of the contradiction. My model does not specify how contradictory information is represented or how it may delay comprehension, but neither does Myers and O'Brien's.

With respect to O'Brien et al.'s (1990) data, in the absence of an interpretation, references to an elaborated concept may be read faster than references to an unelaborated concept because there would be more interpretations matching it and therefore my model would have greater chance to retrieve one of them (provided that it partially matches the focus). An alternative explanation is that an elaborated concept could have been strengthened with repeated use and, thus, according to ACT-R latency equation 2.4, the time needed to retrieve it may be decrease.

10.3 Models for Specific Domains

This section discusses briefly various models or explanations that have been proposed for some of the experimental datasets that were addressed in this dissertation; it also discusses the relationship between this model and the Given-New theory (Haviland & Clark, 1974). I also discuss briefly how the thesis model is connected to models of syntactic processing.

Metaphor-position effects. In their study about how metaphor position affects the reading time of the sentence, Gerrig and Healy (1983) found that subjects are faster to

[5] Whether or not the goal proposition can be a hook for that sentence also depends on how related other words in the sentence are to other propositions and on the sentence position of the object uniquely associated to the goal.

read metaphor-second sentences than to read metaphor-first sentences. They explained this result as caused by subjects making use of "their schematic knowledge of the world to curtail the global process of the literal interpretation. When the metaphors came first, this shortcut was not possible. The extra process of integrating and then abandoning a literal interpretation of the metaphorical phrase was reflected by longer reading time." This explanation essentially coincides with that provided by my model (see Section 4.1): the extra time needed in the metaphor-first sentences is indeed due to an interpretation switch, which is (most of the time) nonexistent in the metaphor-second sentences. For metaphor-second sentences, the model finds the correct interpretation before the metaphor and is able to integrate the metaphor into it.

Moses Illusion. Kamas and Reder's (1995) and Kamas et al.'s (1996)'s account of Moses Illusion is based on a semantic-network representation of knowledge; concepts that are semantically related are connected in this network. For the Moses illusion task, activation spreads from the words occurring in the question to their neighbors in this network; the more input words related to the "correct" interpretation (e.g., the Noah proposition), the more activation is spread towards that node and the higher its activation. If the distorted term is connected to the "correct" interpretation (i.e., they share many semantic features), then the interpretation will receive some spreading activation from the distortion and will be able to achieve an activation level high enough to fool the model into accepting the interpretation.

Again, my model of Moses illusion essentially coincides with Kamas and Reder's model (see Section 4.2): both models fall for good distortions because, given that the similarity between the good distortion and the filler of the corresponding role in the "correct" interpretation is high, the activation spreading to the "correct" proposition is high enough to retrieve it. If the distortion is bad, the similarity between the interpretation and the distortion is small, so the "correct" interpretation sinks under the retrieval threshold. The two models differ in terms of representations: Kamas and Reder seem to favor a more distributed, feature representation of word meanings and the common features, rather than the similarities, drive the spreading of activation[6]. My model also includes a match step, in which the distortion (and the other words in the sentence) are compared to their analogs in the interpretation; the success of the match is also dependent on the activation spreading from the matched term (e.g., the distortion) to the proposition link corresponding to the same role and, thus, ultimately dependent on the similarity between the distortion and the correct term as it is encoded in the interpretation.

[6]This assumption can be embedded in my model too, but rather at a subsymbolic level: what determines similarity may be the number of semantic features shared by two concepts or propositions.

The given–new theory. Haviland and Clark (1974), Clark (1973a) proposed the given–new theory for sentence processing in a discourse context. The given–new theory states that the listener breaks a sentence into new and given information and, then, she attempts to integrate the new information into the text representation extant in memory. In doing so, the listener uses the given information to index the text representation; after finding an antecedent for the given information it attaches the new information to that antecedent. The theory asserts also that, in the absence of given information, the listener can do any of the following: (1) build a bridging inference, (2) construct an entirely new structure corresponding to the new sentence, and (3) attempt to compute again the given and new information.

Haviland and Clark (1974) support their theory with several experimental findings. This theory is mirrored by the behavior of my model in the case of novel sentences embedded in discourse: for these sentences, the model attempts to find a hook, which is a best matching proposition from the context; the model further integrates the novel sentence with the hook proposition. Choosing a hook is an activation-driven process: the given parts of the sentence (when in focus) spread activation to old text propositions and lead to selecting the one that best matches them. That proposition may be eventually rejected, but it serves as a hook for integration with the context. Note that, unlike the given–new theory, my model never builds any bridging inferences; it also never computes the given and the new information in the sentence — it is the activation spreading process that differentiates between the two, because it only happens when given information is in focus.

Syntactic Theories. The model in this dissertation does not perform any syntactic processing: it does not compute parse trees for the input sentence, but rather assumes that thematic roles of the words are already available. Moreover, it only processes sentences with a simple syntactic structure — for instance, it does not deal with multiple-clause sentences. On the other hand, syntactic theories are concerned specifically with determining which word plays what role in the sentence and predicting what types of syntactic structures are difficult for people. Thus, multiple embeddings (e.g., *The man that the cat that the mouse feared scared ran away*) or garden-path sentences (e.g., *The man knows the woman likes green shoes*, *The boat floated sank*) are primary data for such theories — they strive to explain why some of these sentences pose difficulty to people, whereas others are easily understood.

As discussed in Section 9.2, my model predicts that garden-path sentences take extra time, because reassignment of an old role to a new word leads to the rejection of the current interpretation and to the search for a new one. However, my model does not specify how and whether thematic roles are reassigned after the current interpretation is rejected. For instance, let us assume that for the input words *The man hit*, the model selected an interpretation in which *man* was the agent. If later, when the words *by the ball* are input, *ball* is recognized as an agent, the model needs to reject the current interpretation and search for an interpretation involving the words *man, hit, ball* and in which *ball* is the agent. If it retrieves one such proposition that, additionally, has the properties that *man* is its patient and *hit* is its verb, then the model is lucky as it does not need to perform any other kind of syntactic repair. The chance of retrieving the proposition with the correct role assignment depends on factors such as the frequency of that role assignment in the language and on the existence of propositions with alternative role assignments. Hence, with respect to predicting the difficulty of sentences involving syntactic ambiguities, the model's take is that the infrequent assignment may be hard to solve. For instance, *The boat floated sank* can be difficult because there may be no proposition in the background knowledge involving the the words *boat, floated, sank* and in which *sank* is the verb. Interestingly, the model predicts that if more words are interposed between *floated* and *sank*, the difficulty may wash away — that is, *The boat floated on a rainy day on Mississippi sank* may be easier because *floated* disappears from focus when the disambiguation *sank* is read and, thus, does not actually contribute to the search for an interpretation[7].

Although we can speculate on the syntactic implications of the model in this dissertation, fundamentally, it addresses questions about language comprehension that are different from those of syntactic theories. A comparison between my model and syntactic theories can be only very general, as they cannot be tested on similar tasks (e.g., my model works only on single-clause sentences and does not parse separately components of word phrases). Nonetheless, it may be worthwhile to point out similarities in the reasoning processes that underline them. Henceforth, I refer to Lewis' (1993) theory of syntactic processing (Lewis, 1993, 1998), which has been embodied in a computational model based on the Soar cognitive architecture (Newell, 1990; Laird, Newell, & Rosenbloom, 1987). One conclusion that both Lewis (1998) and my model share is that search (for a correct parse tree or for the right semantic interpretation, respectively) must be limited: it is unfeasible to explore the entire space of possibilities. Thus, Lewis proposes limited-repair parsing, a technique in which, when a syntactic ambiguity arises, rather than backtracking to other parsing options, the parser instantly repairs the current syntax tree by using a special operator (called *snip*). On the other hand, when an interpretation proves fallacious, my model moves to another one, without actually checking (i.e., backtracking) that all previous words match the new interpretation[8]. With respect to the role of the frequency of the syntactic construct in the easiness of disambiguation, Lewis' (1998) model does not seem to rely on it; the difficulty of a sentence depends on whether the snip operator can be applied or not. Thus,

[7]Note, however, that in this situation, the model would not find a perfectly matching interpretation for the input sentence, but just a partially matching interpretation with which it would be satisfied. Thus, the comprehension would be poorer.)

[8]When searching for a new interpretation, my model is informed only by the previous three words; moreover, it does not attempt to check whether those words actually occur in the new interpretation in the same thematic roles.

because it cannot be applied for any of the sentences *The boat floated on a rainy day on Mississippi sank* or *The boat floated sank*, Lewis' model predicts that these sentences are equally difficult.

Lately, Lewis (1999a) has produced an ACT-R model of syntactic processing. This model is activation driven — it posits that human syntactic processing is governed by activation spreading, activation decay and base-level activation (reflecting the frequency of various categories — for example, how often *make* is a verb versus a noun). Lewis' (1999a) model predicts that the difficulty of a syntactic structure is given by the interference between various candidates to the same thematic role (e.g., in the sentence *The man that the cat that the mouse feared scared ran away*, there are three nouns competing for the agent role of the verb *scared*). Interestingly, for both Lewis' (1999a) model and for mine, activation spreading and focus are ACT-R features that play a central role. However, unlike in my model, in Lewis' (1999a) model, associations (which determine spreading of activation) are used in an inhibitory rather than facilitatory way: the more candidates to the same role, the fewer activation spread to each of them and the more difficult the processing. In my model, associations are redefined to reflect semantic similarity and, thus, are not subjected to this fan effect; they serve the function of selecting those propositions that contain concepts similar to those in focus.

Chapter 11

Conclusions

In this chapter I conclude this dissertation by summarizing its contributions (Section 11.1), pointing out its limitations (Section 11.2), and drawing some directions for future work (Section 11.3).

11.1 Contributions

This dissertation presented a computational model of sentence processing, implemented in ACT-R. The principal virtue of this model is that it uses a simple mechanism to explain several behavioral data from different domains. The basic mechanism is a greedy search-and-match process, in which, with each input word processed, a matching long-term–memory interpretation is sought. The search is guided by the last few words read: these words are kept in the focus of attention and spread activation to the propositions in memory, raising the overall activation of the interpretations that contain them. The match is very loose, in the sense that items that are not identical but similar enough can match each other; successful matching is implemented as successful retrieval. The process of comprehension produces a propositional structure for the input sentence and also records local comprehension failures; the propositional structure contains pointers to the long-term-memory interpretation found by the model during comprehension.

Using this mechanism, I was able to explain phenomena such as metaphor-position effects on comprehension (Gerrig & Healy, 1983) and Moses illusion (Reder & Kusbit, 1991; Ayers et al., 1996). By restricting the long-term memory to propositions belonging to text context, I could also account for verification and comprehension of metaphoric sentences embedded in discourse (Budiu & Anderson, 2001, 2000a). I also showed that the propositional structures created during comprehension can explain the pattern of intrusions made by people when they need to recall sentences that they had studied before (Bower et al., 1979). In all these cases, the model simulated successfully the latency and accuracy patterns produced by human subjects.

This dissertation argued that the model proposed presents a (relatively) scalable solution to the problem of comprehension: its performance does not depend significantly on the size of the knowledge base used. It also compares well with other models of sentence processing (e.g., Kintsch, 2000) on tasks that were chosen to test those models.

Beside describing a model of sentence comprehension, I also discussed the choices that had to be made in the design of such a model. We saw that, within a production-system architecture such as ACT-R, to achieve the speed of human comprehension, a sentence-processing model must rely heavily on activation-spreading from words to propositions. To account for the flexibility of comprehension, the spreading of activation must be based on semantic similarity. I also showed that a fragmented propositional representation is better suited to the task of modeling sentence processing than an atomic propositional representation.

In his (1988) paper, Kintsch argued that modeling comprehension with a top-down, rule-based system is not a realistic task, because "it is difficult to design a production system powerful enough to yield the right results but flexible enough to work in an environment characterized by almost infinite variability." In this dissertation I showed that, as Kintsch (1988) claims, a model that performs an exclusively procedural search is not viable. On the other hand, a simple production system with a powerful, similarity-based mechanism of spreading activation may offer the right mixture of bottom-up and top-down processes: spreading activation may lead to the selection of the right interpretation and, once that interpretation selected, it can ensure flexibility of comprehension through a flexible matching mechanism.

11.2 Limitations

The sentence-processing model that I propose makes a number of important simplifying assumptions.

Syntactic processing. The model does not account for syntactic processing and it assumes that syntactic and semantic processes are independent. Thus, whereas it produces reading times comparable with those of people, it is not clear whether this speed can be maintained when syntactic processes are added to the model.

Processing of word phrases. We saw that the model treats noun or verb phrases as single concepts and does not process separately component words (i.e., it treats *drops of molten silver* as a concept with extant meaning); also, it does not explain how the meaning of such a complex concept is formed, but rather takes it for granted. In Section 4.1 we saw that it is possible to extend the model to act recursively on such noun phrases; however, this extension is only at the level of a verbal theory.

Processing of multiple-clause sentences. My model works only with one-clause sentences; nonetheless, in normal comprehension, multiple-clause sentences are quite common. Again, one may expect that the model be applied recursively to such sentences, but building a coherent propositional structure for them may be more challenging (for instance, in a sentence such as *The man that Jim saw was tall*, the relative pronoun *that* must be bound to *the man* to produce a complete propositional representation for the relative clause).

Thematic-role cues. In selecting a current interpretation, my model uses no thematic-role information; only at matching does it check whether the candidate interpretation has a similar thematic role as the current word. It is hard to assess whether or not thematic roles can be ignored when searching for an interpretation; if two or three words can combine together in very few propositions, then it may make sense to ignore thematic roles. On the other hand, when understanding text, people may limit (as the model does) their set of candidate interpretations to those coming from that text; thus, two or three words may have a greater chance to identify uniquely a discourse proposition than to identify a proposition from the entire long-term memory. Also, in tasks involving verification of isolated sentences, it often happens that a few words uniquely identify the proposition (e.g., there are not many propositions involving *Noah, ark,* and *animals* or *Bible, whale* and *Jonah*).

Rudimentary discourse processing. Although the model proposed successfully simulates comprehension of text, it does not perform all the functions normally associated to text processing; thus, this model does not resolve pronominal references, is not concerned with binding multiple references to the same object, does not monitor goals, and does not make complex inferences (such as detection of contradictions) or elaborations on text. It is true that none of these were necessary to explain the set of behavioral data modeled, but, nonetheless, they are part of text processing and occur in other circumstances.

Relationship between background and discourse knowledge. When simulating discourse processing, the model searches for an interpretation among text propositions, excluding long-term–memory propositions from the set of candidates. This is sensible if we think that, when processing text, we are concerned with relating a new sentence to those that we read before. However, this would mean that we could read nonsensical sentences involving concepts that have antecedents in the story and not notice any problem. For instance, a passage such as *Joe enjoyed spending his evening by the fire. He liked to hold his cat in his lap. Sometimes he read humor books loudly and petted the cat gently on her head* may be followed by the sentence *The cat gently petted the fire*; there would be no interpretation for this sentence in the discourse, but the model would find a hook to which it would relate it. At no point would the model detect any nonsense because it would not check background knowledge. The reason why the model does not check background knowledge after failing to find a text interpretation for a sentence is that sometimes that failure is relevant (for instance, in the case of metaphoric sentences such as *The hens clucked noisily*, following a women's meeting passage; in that case, failure of finding an interpretation represents failure of comprehending the metaphor).

11.3 Future Work

I plan to use this model to account for other empirical phenomena such as word priming in texts or lexical-ambiguity resolution. I also intend to simulate text-inferencing studies such as O'Brien et al. (1990), Albrecht and Myers (1995), as it would be interesting to see what kinds of text inferences could be accounted for with limited mechanisms such as activation spreading or to the structure of background knowledge.

Another immediate direction of research is implementing my model in ACT-R 5.0. ACT-R 5.0 is a new ACT-R version that departs significantly from the ACT-R 4.0 architecture (Anderson & Lebiere, 1998) used for my sentence-processing model. I hope that the process of transfer from one variant of the architecture to another gives interesting insights on those ACT-R assumptions that are crucial for the success of my model.

Eliminating some of the limitations enumerated in the previous sections can be a subject for future work. One of the most important investigations still to be made is related to whether people use thematic role information to select interpretations. If that is the case, the model should be modified to allow directed priming, from words in the goal only to those propositions containing the same words in the same thematic roles. Although in the sentence-database simulation the model was successful in spite of not using thematic-role cues, that database may still be not representative enough for real language (as it contains only noun–verb–noun sentences).

Other directions of future work consist of testing some of the predictions made by the model. For instance, the model predicts that, if a word appears in few propositions, then placing it at the beginning of the sentence will result in faster comprehension than placing it at the end, because it will allow an early selection of the correct interpretation and, thus, will avoid failures and retries. Similarly, according to this model, comprehending sentences with few overlapping contexts should be faster than comprehending sentences with many overlapping contexts (for instance, out of discourse context, the sentence *Noah took the animals on the ark* should be faster than *In United States, people love freedom/money/children/athletes/hamburgers*). Another prediction concerns the Moses illusion: in Section 4.2 we saw that the model predicts that distortions at the end of the sentence are harder to detect than distortions at the beginning, provided that there are no overlaps between the facts involving the distortion (e.g., *Moses parted the Red sea*) and the correct, undistorted interpretation (e.g., *Noah took the animals on the ark*).

Index

References

Albrecht, J., & Myers, J. (1995). Role of context in accessing distant information during reading. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *21*, 1459-1468.

Albrecht, J., & Myers, J. (1998). Accessing distant text information during reading: effects of contextual cues. *Discourse Processes*, *26*, 87-107.

Altmann, G., & Steedman, M. (1988). Interaction with context during human sentence processing. *Cognition*, *30*, 191-238.

Anderson, J. (1972). *A stochastic model of sentence memory.* Unpublished doctoral dissertation, Stanford University.

Anderson, J., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, *38*, 341-380.

Anderson, J., Budiu, R., & Reder, L. (in press). *A theory of sentence memory as part of a general theory of memory.* Journal of Memory and Language.

Anderson, J., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, New Jersey: Lawrence Erlbaum Associates Publishers.

Anderson, R., & Ortony, A. (1975). On putting apples into bottles: A problem of polysemy. *Cognitive Psychology*, *7*, 167-180.

Ayers, M., Reder, L., & Anderson, J. (1996). *Accepting false information now and believing it later: Partial matching and false information in the Moses illusion.* Unpublished manuscript.

Bartlett, F. (1932). *Remembering: A study in experimental and social psychology.* New York & London: Cambridge University Press.

Barton, S., & Sanford, A. (1993). A case study of anomaly detection: Shallow semantic processing and cohesion establishment. *Memory and Cognition*, *21*, 477-487.

Black, M. (1962). *Models and metaphors.* Ithaca, NY: Cornell University Press.

Black, M. (1979). More about metaphor. In A. Ortony (Ed.), *Metaphor and thought.* Cambridge University Press.

Blasko, D., & Connine, C. (1993). Effects of familiarity and aptness on metaphor processing. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *19*, 295-308.

Bower, G., Black, J., & Turner, T. (1979). Scripts in memory for texts. *Cognitive Psychology*, *11*, 177-220.

Bradshaw, G., & Anderson, J. (1982). Elaborative encoding as an explanation of levels of processing. *Journal of Verbal Learning and Verbal Behavior*, *21*, 165-174.

Bransford, J., & Johnson, M. (1972). Contextual prerequisites for understanding: Some investigations of comprehension and recall. *Journal of Verbal Learning and Verbal Behavior*, *11*, 717-726.

Budiu, R., & Anderson, J. (2000a). *Comprehending anaphoric metaphors.* Submitted for publication.

Budiu, R., & Anderson, J. (2000b). Integration of background knowledge in sentence processing: a unified theoy of metaphor understanding, semantic illusions and text memory. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the third international conference on cognitive modeling* (p. 50-57). Netherlands: Universal Press.

Budiu, R., & Anderson, J. (2001). *Word learning in context: Metaphors and neologisms* (Tech. Rep. No. CMU-CS-01-147). Pittsburgh, PA: School of Computer Science, Carnegie Mellon University.

Cacciari, C., & Glucksberg, S. (1992). Understanding figurative language. In M. Gernsbacher (Ed.), *Handbook of psycholinguistics.* San Diego, CA.

Carnine, D., Kameenui, E. J., & Coyle, G. (1984). Utilization of contextual information in determining the meaning of unfamiliar words. *Reading Research Quaterly*, *19*, 188–204.

Clark, H. (1973a). Comprehension and the given–new contract. In *Conference on the role of grammar in interdisciplinary linguistic research.* Bielefeld, Germany.

Clark, H. (1973b). The language–as–fixed-effect fallacy: A critique of language statistics in psychological research. *Journal of Verbal Learning and Verbal Behavior*, *12*, 335-359.

Clark, H., & Haviland, S. (1974). Psychological processes as linguistic explanation. In D. Cohen (Ed.), *Explaining linguistic phenomena.* Washington: V.H. Winston.

Clark, R., & Gibson, E. (1988). A parallel model for adult sentence processing. In *Proceedings of the 10th annual conference of the Cognitive Science Society* (p. 270-277). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Cook, A., Halleran, J., & O'Brien, E. (1998). What is readily available during reading? A memory-based view of text processing. *Discourse Processes*, *26*, 109-129.

Dascal, M. (1987). Defending literal meaning. *Cognitive Science*, *11*, 259-281.

Duffy, S., Henderson, J., & Morris, R. (1989). Semantic facilitation of lexical access during sentence processing. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *15*, 791-801.

Duffy, S., Morris, R., & Rayner, K. (1988). Lexical ambiguities and fixation times in reading. *Journal of Memory and Language*, *27*, 429-446.

Erickson, T., & Mattson, M. (1981). From words to meaning: A semantic illusion. *Journal of Verbal Learning and Verbal Behavior*, *20*, 540-552.

Fischer, U. (1994). Learning words from context and dictionaries: An experimental comparison. *Applied Psycholinguistics*, *15*, 551–574.

Foltz, P., Kintsch, W., & Landauer, T. (1998). The measurement of textual coherence with Latent Semantic Analysis. *Discourse Processes*.

Foss, D. (1988). Experimental psycholinguistics. *Annual review of Psychology*, *39*, 301-348.

Francis, W., & Kucera, H. (1982). *Frequency analysis of English usage: Lexicon and grammar.* Boston: Houghton Mifflin.

Garrod, S., Freudenthal, D., & Boyle, E. (1994). The role of different types of anaphor in the on-line resolution of sentences in a discourse. *Journal of Memory and Language*, *33*, 39-68.

Gentner, D., & France, I. (1988). The verb mutability effect: Studies of the combinatorial semantics of nouns and verbs. In S. Small, G. Cottrell, & M. Tanenhaus (Eds.), *Lexical ambiguity resolutions: Perspectives from psycholinguistics, neuropsychology and artificial intelligence* (p. 343-382). Morgan Kauffman.

Gernsbacher, M., & Keysar, B. (1995, November). *The role of supression in metaphor interpretation.* Paper presented at the 36th Annual Meeting of the Psychonomic Society, Los Angeles.

Gerrig, R. (1989). The time course of sense creation. *Memory and Cognition*, *17*, 197–207.

Gerrig, R., & Healy, A. (1983). Dual processes in metaphor understanding: Comprehension and appreciation. *Journal of Experimental Psychology: Memory and Cognition*, *9*, 667–675.

Gerrig, R., & McKoon, G. (1998). The readiness is all: The functionality of memory-based text processing. *Discourse Processes*, *26*, 67-86.

Gibbs, R. (1990). Comprehending figurative referential descriptions. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *16*, 56–66.

Gibson, E. (1991). *A computational theory of human linguistic processing: Memory limitations and processing breakdown.* Unpublished doctoral dissertation, Carnegie Mellon University. (Available as technical report CMU-CMT-91-125.)

Giora, R. (1997). Understanding figurative and literal language: The graded salience hypothesis. *Cognitive Linguistics*, *8*, 183-206.

Glucksberg, S., & Estes, Z. (2000). Feature accessibility in conceptual combination: Effects of context induced relevance. *Psychonomic Bulletin and Review*, *7*, 510-515.

Glucksberg, S., Glidea, P., & Bookin, H. (1982). On understanding literal speech: Can people ignore metaphors? *Journal of Verbal Learning and Verbal Behavior*, *21*, 85–98.

Glucksberg, S., & Keysar, B. (1990). Understanding metaphorical comparisons: Beyond similarity. *Psychological Review*, *97*, 3-18.

Glucksberg, S., McGlone, M., & Manfredini, D. (1997). Property attribution in metaphor comprehension. *Journal of Memory and Language*, *36*, 50-76.

Goldvarg, Y., & Glucksberg, S. (1998). Conceptual combinations: The role of similarity. *Metaphor and Symbol.*

Green, D. W., Mitchell, D. C., & Hammond, E. J. (1981). The scheduling of text integration processes in reading. *Quarterly Journal of Experimental Psychology: Human Experimental Psychology*, *33A*, 455-464.

Haviland, S., & Clark, H. (1974). What's new? Aquiring new information as a process in comprehension. *Journal of Verbal Learning and Verbal Behavior*, *13*, 512-521.

Herman, P., Anderson, R., Pearson, P., & Nagy, W. E. (1987). Incidental acquisition of word meaning from expositions with varied text features. *Reading Research Quaterly*, *22*, 263–284.

Inhoff, A., Lima, S., & Carroll, P. (1984). Contextual effects on metaphor comprehension in reading. *Memory and Cognition*, *2*, 558-567.

Jaarsveld, H., Dijkstra, T., & Hermans, D. (1997). The detection of semantic illusions: Task specific effects for similarity and position of distorted terms. *Psychological research, 59*, 219-230.

Janus, R., & Bever, T. (1985). Processing of metaphoric language: An investigation of the three-stage model of metaphor comprehension. *Journal of Psycholinguistic Research, 14*, 473-487.

Just, M., & Carpenter, P. (1980). A theory of reading: From eye fixations to comprehension. *Psychological Review, 87*, 329-354.

Kamas, E., & Reder, L. (1995). The role of familiarity in cognitive processing. In E. O'Brien & R. Lorch (Eds.), *Sources of coherence in reading* (p. 177-202). Hillsdale,NJ: Erlbaum.

Kamas, E., Reder, L., & Ayers, M. (1996). Partial matching in the Moses illusion: Response bias not sensitivity. *Memory and cognition, 24*, 687-699.

Keenan, J., & Jennings, T. (1995). The role of word-based priming in inference research. In R. Lorch & E. O'Brien (Eds.), *Sources of coherence in reading.* Hillsdale, New Jersey: Erlbaum.

Kellas, G., Paul, S., Martin, M., & Simpson, G. (1991). Contextual feature activation and meaning access. In G. Simpson (Ed.), *Understanding word and sentence* (p. 47-71). Amsterdam: North Holland.

Kennison, S., & Gordon, P. (1997). Comprehending referential expressions during reading: Evidence from eyetracking. *Discourse Processes, 24*, 229-252.

Keysar, B. (1989). On the functional equivalence of literal and metaphorical interpretations in discourse. *Journal of Memory and Language, 28*, 375-385.

Keysar, B. (1994). Discourse context effects: Metaphorical and literal interpretations. *Discourse Processes, 18*, 247-269.

Keysar, B., Shen, Y., Glucksberg, S., & Horton, W. (2000). Conventional language: How metaphorical is it? *Journal of Memory and Language*(43), 576-593.

Kintsch, W. (1988). The use of knowledge in discourse processing: A construction-integration model. *Psychological Review, 95*, 163-182.

Kintsch, W. (2000). Metaphor comprehension: A computational theory. *Psychonomic Bulletin and Review, 7*, 257-266.

Kintsch, W. (2001). Predication. *Cognitive Science, 25*, 173-202.

Kintsch, W., & van Dijk, T. (1978). Toward a model of text comprehension and production. *Psychological Review*, *85*, 363-394.

Laham, D. (1997). Latent Semantic Analysis approaches to categorization. In M. Shafto & M. Johnson (Eds.), *Proceedings of the 19th annual conference of the cognitive science society.* Mahwah, NJ: Erlbaum.

Laird, J., Newell, A., & Rosenbloom, P. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1-64.

Lakoff, G. (1987). *Women, fire, and dangerous things.* Chicago, Il: University of Chicago Press.

Lakoff, G., & Johnson, M. (1990). *Metaphors we live by.* Chicago, IL: University of Chicago Press.

Landauer, T., Laham, D., Redher, R., & Schreiner, M. (1997). How well can a passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In M. Shafto & M. Johnson (Eds.), *Proceedings of the 19th annual conference of the cognitive science society.* Mahwah, NJ: Erlbaum.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, *105*, 221–240.

Lewis, R. (1993). *An architecturally-based theory of human sentence comprehension.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA. (Available as technical report CMU-CS-93-226.)

Lewis, R. (1998). Reanalysis and limited repair parsing: Leaping off the gardenpath. In J. Fodor & F. Ferreira (Eds.), *Reanalysis in sentence processing* (p. 247-285). Netherlands: Kluwer Academic Publishers.

Lewis, R. (1999a, March). *Attachment without competition: A race-based model of ambiguity resolution in a limited working memory.* Presented at the CUNY Sentence Processing Conference, New York.

Lewis, R. (1999b). Specifying architectures for language processing: Process, control and memory in parsing and interpretation. In M. Crocker, M. Pickering, & C. Clifton (Eds.), *Architectures and mechanisms for language processing.* Cambridge University Press.

Long, D., & Golding, J. (1993). Superordinate goal inferences: Are they automatically generated during comprehension? *Discourse Processes*, *16*, 55-73.

Lorch, J., R.F. (1998). Memory-based text processing: Assumptions and issues. *Discourse Processes.*

Lovett, M. (1998). Choice. In J. Anderson & C. Lebiere (Eds.), *The atomic components of thought.* Lawrence Erlbaum Associates.

Lucas, M. (1999). Context effects in lexical access: A meta-analysis. *Memory and Cognition*, *27*, 385-398.

MacDonald, M., Pearlmutter, N., & Seidenberg, M. (1994). Lexical nature of syntactic ambiguity resolution. *Psychological Review*, *101*, 676-703.

Marslen-Wilson, W. (1973). Linguistic structure and speech shadowing at very short latencies. *Nature*, *224*, 522-523.

Marslen-Wilson, W. (1975). Sentence perception as an interactive parallel process. *Science*, 226-228.

Masson, M., & MacLeod, C. (2000). Taking the "Text" out of context effects in repetition priming of word identification. *Memory and Cognition*, *28*, 1090-1097.

Matthiessen, F. (Ed.). (1950). *The Oxford book of American verse.* New York, NY: Oxford University Press.

McClelland, J., & Rumelhart, D. (1981). An interactive model of context effects in letter perception: I. An account of basic findings. *Psychological Review*, *88*, 375-407.

McKeown, M. (1985). The acquisition of word meaning from context by children of high and low ability. *Reading Research Quaterly*, *20*, 482–496.

McKoon, G., & Ratcliff, R. (1992). Inference during reading. *Psychological Review*, *99*, 440-466.

McRae, K., Spivey-Knowlton, M., & Tanenhaus, M. (1998). Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, *38*, 283-312.

Mitchell, D., & Green, D. (1978). The effects of context and content on immediate processing in reading. *Quaterly Journal of Experimental Psychology*, *28*, 325-337.

Murray, J., Klin, C., & Myers, J. (1993). Forward inferences in narrative text. *JML*, *32*, 464-473.

Myers, J., & O'Brien, E. (1998). Accessing the discourse representation during reading. *Discourse Processes*, *26*, 131-157.

Nagy, W., Anderson, R., & Herman, P. (1987). Learning word meanings from context during normal reading. *American Educational Research Journal*, *24*, 237–270.

Nagy, W., & Genter, D. (1990). Semantic constraints on lexical categories. *Language and Cognition Processes*, *5*, 169–201.

Nagy, W., Herman, P., & Anderson, R. (1985). Learning words from context. *Reading Research Quaterly*, *20*, 233–253.

Nagy, W., & Scott, J. A. (1990). Word schemas: Expectations about the form and meaning of new words. *Cognition and Instructions*, *7*, 105-127.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Noordman, L., & Vonk, W. (1998). Memory based processing in understanding causal information. *Discourse Processes.*

Oakhill, J., Garnham, J., & Vonk, W. (1989). The on-line construction of discourse models. *Language and Cognitive Processes*, *4*, SI 236-386.

O'Brien, E., & Myers, J. (1985). When comprehension difficulty improves memory for text. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *11*, 12-21.

O'Brien, J., E., Plewes, S., & Albrecht, J. (1990). Antecedent retrieval processes. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *16*, 241-249.

Onifer, W., & Swinney, D. (1981). Accessing lexical ambiguity during sentence comprehension: Effects of frequency of meaning and contextual bias. *Memory and Cognition*, *9*, 225–236.

Onishi, K., & Murphy, G. (1993). Metaphoric reference: When metaphors are not understood as easily as literal comprehension. *Memory and Cognition*, *21*, 763-772.

Ortony, A., Schallert, D., Reynolds, R., & Antos, S. (1978). Interpreting metaphors and idioms: Some effects on comprehension. *Journal of Verbal Learning and Verbal Behavior*, *17*, 465–477.

Ortony, A., Vondruska, R., Foss, M. A., & Jones, L. (1985). Salience, similes and the asymmetry of similarity. *Journal of Memory and Language*, *24*, 569-594.

Owens, J., Bower, G., & Black, J. (1979). The "soap opera" effect in story recall. *Memory and Cognition*, *7*, 185-191.

Paul, S., Kellas, G., & Juola, J. (1992). Priming effects on lexical accesses and decision processes: A signal detection analysis. *Psychological Research*, *54*, 202-211.

Paul, S., Kellas, G., Martin, M., & Clark, M. (1992). The influence of contextual features on the activation of ambiguous word meanings. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *18*, 703-717.

Potter, M., & Lombardi, L. (1990). Regeneration in the short term recall of sentences. *Journal of Memory and Language*, *19*, 633-654.

Quiller-Couch, A. (Ed.). (1940). *The Oxford book of English verse* (New York, NY ed.). Oxford University Press.

Rayner, K., & Duffy, S. (1986). Lexical complexity and fixation times in reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory and Cognition*, *14*, 191-201.

Rayner, K., & Frazier, L. (1989). Selection mechanisms in reading lexically ambiguous words. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *15*, 779-790.

Rayner, K., & Morris, R. (1991). Comprehension processes in reading ambiguous sentences: Reflections from eye movements. In G. Simpson (Ed.), *Understanding word and sentence.* Amsterdam: North-Holland.

Reddy, M. (1993). The conduit metaphor. In A. Ortony (Ed.), *Metaphor and thought.* Cambridge, MA: Cambridge University Press.

Reder, L. (1980). The role of elaboration in the comprehension and retention of prose: A critical review. *Review of Educational Research*, *50*, 5–53.

Reder, L. (1983). What kind of pitcher can a catcher fill? Effects of priming in sentence comprehension. *Journal of Verbal Learning and Verbal Behavior*, *22*, 189-202.

Reder, L., & Kusbit, G. (1991). Locus of the Moses illusion: Imperfect encoding, retrieval, or match? *Journal of Memory and Language*, *30*, 385-406.

Salvucci, D., & Anderson, J. (2001). Integrating analogical mapping and general problem solving: The path-mapping theory. *Cognitive Science*, *25*, 67-110.

Sanford, A. (1990). On the nature of text driven inference. In D. Balota, G. Flores D'Arcais, & K. Rayner (Eds.), *Comprehension processes in reading* (p. 515-535). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Sanford, A., & Garrod, S. (1998). The role of scenario mapping in text comprehension. *Discourse processes*, *26*, 159-190.

Schank, R. (1982). *Dynamic memory: A theory of reminding and learning in computers and people.* Cambridge: Cambridge University Press.

Schank, R. (1999). *Dynamic memory revisited.* Cambridge: Cambridge Universit Press.

Schank, R., & Abelson, R. (1977). *Scripts, plans, goals and understanding.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Schustack, M., & Anderson, J. (1979). Effects of analogy to prior knowledge on memory for new information. *Journal of Verbal Learning and Verbal Behavior*, *18*, 565-583.

Searle, J. (1979). Metaphor. In A. Ortony (Ed.), *Metaphor and thought.* Cambridge University Press.

Shefelbine, J. (1990). Student factors related to variability in learning word meanings in context. *Journal of Reading Behavior*, *22*, 71.

Shinjo, M., & Myers, J. (1987). The role of context in metaphor comprehension. *Journal of Memory and Language*, *26*, 226-241.

Simpson, G. (1994). Context and the processing of ambiguous words. In M. Gernsbacher (Ed.), *Handbook of psycholinguistics* (p. 359-374). San Diego, CA: Academic Press.

Stein, B., & Bransford, J. (1979). Constraints on effects of elaboration: Effects of precision and subject generation. *Journal of Verbal Learning and Verbal Behavior*, *18*, 769-777.

Sternberg, R., & Powell, J. (1983). Comprehending verbal comprehension. *American Psychologist*, *38*, 878–893.

Swinney, D. (1979). Lexical access during sentence comprehension: (Re)Consideration of context effects. *Journal of Verbal Learning and Verbal Behavior*, *18*, 645-659.

Tabossi, P. (1988). Accessing lexical ambiguities in different types of sentential context. *Journal of Memory and Language*, *17*, 324-340.

Tabossi, P. (1989). What's in a context? In D. Gorfein (Ed.), *Resolving semantic ambiguity* (p. 25-39). New York: Springer Verlag.

Tabossi, P. (1991). Understanding words in context. In G. Simpson (Ed.), *Understanding word and sentence* (p. 1-22). Amsterdam: North-Holland.

Tourangeau, R., & Rips, L. (1991). Interpreting and evaluating metaphors. *Journal of Memory and Language*, *30*, 452-472.

Tourangeau, R., & Sternberg, R. (1981). Aptness in metaphor. *Cognitive Psychology*, *13*, 27-55.

Townsend, J. (1974). Issues and models concerning the processing of a finite number of inputs. In B. Kantowitz (Ed.), *Human information processing: Tutorials in performance and cognition* (p. 133-186). Hilsdale, NJ: Lawrence Erlbaum Associates.

Traxler, M., Bybee, M., & Pickering, M. (1997). Influence of connectives on language comprehension: Eye-tracking evidence for incremental interpretation. *The Quaterly Journal of Experimental Psychology, 50A*, 481-497.

Tyler, L., & Marslen-Wilson, W. (1982). The resolution of discourse anaphors: Some online studies. *Text, 2*, 263-291.

Ullmann, S. (1966). *Language and style.* New York: Barnes and Noble, Inc.

van Daalen–Kapteijns, M. M., & Elshout-Mohr, M. (1981). The acquisition of word meanings as a cognitive learning process. *Journal of Verbal Learning and Verbal Behavior, 20*, 286–399.

van Oostendorp, H., & de Mul, S. (1990). Moses beats Adam: A semantic relatedness effect on a semantic illusion. *Acta Psychologica, 74*, 35-46.

van Oostendorp, H., & Kok, I. (1990). Failing to notice errors in sentences. *Language and cognitive processes, 5*, 105-113.

Webster's 9th new collegiate dictionary. (1991). Springfield, MA: Merriam-Webster.

Werner, H., & Kaplan, E. (1950a). The acquisition of word meanings: A developmental study. *Monographs of the Society for Research in Child Development, 15*(51).

Werner, H., & Kaplan, E. (1950b). Development of word meaning through verbal context: An experimental study. *Journal of Psychology, 29*, 251–257.

Wolfe, M., Schreiner, M., Rehder, R., Laham, D., Foltz, P., T.K., L., & Kintsch, W. (1998). Learning from text: Matching reader and text by Latent Semantic Analysis. *Discourse Processes, 25*, 309-336.

Xiaolong, L. (1988). Effects of contextual cues on inferring and remembering meanings of words. *Applied Linguistics, 9*, 402–413.

Zwaan, R., & Radvansky, G. (1998). Situation models in language comprehension and memory. *Psychological Bulletin, 123*, 162-185.