Teaching Machines to Classify from Natural Language Interactions

Shashank Srivastava

September 2018
CMU-ML-18-109

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Tom Mitchell, Chair
Taylor Berg-Kirkpatrick
William Cohen
Dan Roth (University of Pennsylvania)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by: Defense Advanced Research Projects Agency award numbers FA875009C0172 and W911NF1210034; National Science Foundation award number IIS1304939; Air Force Research Laboratory award numbers FA87501220342, FA87501320005 and FA865013C7360; US ARMY Research Office award number W911NF1410436; a gift from Yahoo/Verizon; and a grant from Samsung Electronics.

Keywords: Machine learning, Natural Language Understanding, Semantic Parsing, Conversational Learning, Grounded Language, Computational Linguistics, Pragmatics, Concept Learning

"Die Grenzen meiner Sprache bedeuten die Grenzen meiner Welt."
(The limits of my language mean the limits of my world.)

– Ludwig Wittgenstein
Tractatus Logico-Philosophicus

Abstract

Humans routinely learn new concepts using natural language communications, even in scenarios with limited or no labeled examples. For example, a human can learn the concept of a phishing email from natural language explanations such as 'phishing emails often request your bank account number'. On the other hand, purely inductive learning systems typically require a large collection of labeled data for learning such a concept. We believe that advances in Computational Linguistics and the growing ubiquity of computing devices together can enable people to teach computers classification tasks using natural language interactions.

Learning from language presents some key challenges. A preliminary challenge lies in the basic problem of *learning to interpret* language, which refers to an agent's ability to map natural language explanations in pedagogical contexts to formal semantic representations that computers can process and reason over. A second challenge is that of *learning from interpretations*, which refers to the mechanisms through which interpretations of language statements can be used by computers to solve learning tasks in the environment. We address aspects of both these problems, and provide an interface for guiding concept learning methods using language.

For *learning from interpretation*, we focus on concept learning (binary classification) tasks. We demonstrate that language can define rich and expressive features for learning tasks, and show that machine learning can benefit substantially from this ability. We also investigate assimilation of linguistic cues in everyday language that implicitly provide constraints for classification models (e.g., 'Most emails are not phishing emails'). In particular, we focus on conditional statements and linguistic quantifiers (such as usually, never, etc.), and show that such advice can be used to train classifiers even with few or no labeled examples of a concept.

For *learning to interpret*, we develop new algorithms for semantic parsing that incorporate pragmatic cues, including conversational history and sensory observation, to improve automatic language interpretation. We show that environmental context can enrich semantic parsing methods by not only providing discriminative features, but also reducing the need for expensive labeled data used for training them.

A separate but immensely valuable attribute of human language is that it is inherently conversational and interactive. We also briefly explore the possibility of agents that can *learn to interact* with a human teacher in a mixed-initiative setting, where the learner can also proactively engage the teacher by asking questions, rather than only passively listen. We develop a reinforce learning framework for learning effective question asking strategies in context of conversational concept learning.

Acknowledgments

For your guidance and mentorship, thank you Tom. You spurred me towards exploring learning from grounded language, curiously when my research most needed grounding. But thank you most for your kindness and patience while I found my path, and for your exceptional enthusiasm for research.

For your generosity and infectious love of language, thank you Ed. I specially treasure my time working with you, and your influence has made its way into the fabric of this thesis in multiple ways.

For your gift of time and support, thank you Dan. You have been a sounding board for nascent thoughts and a well of ideas whenever we spoke. I found both friendship and structure in my time away from Pittsburgh through the CogComp group.

Other members of my thesis committee – William Cohen and Taylor Berg-Kirkpatrick have greatly improved this dissertation, both by bringing new and valuable perspectives to my awareness, as well as through suggestions on improving the technical presentation.

My research has benefitted from discussions with several members of the NELL group: Amos Azaria, Anthony Platanios, Abulhair Saparov, Ndapa Nakashole, Derry Wijaya, Jayant Krishnamurthy, Bishan Yang and Bryan Kisiel. Igor Labutov has been an especially close friend, colleague and collaborator in the work described in this dissertation. I have also learnt from some excellent teachers at CMU. I would like to especially mention Ryan Tibshirani, Larry Wasserman, Noah Smith, Chris Dyer, Ed Hovy and Roni Rosenfeld. Your love of teaching and respect for the student has been joyful to behold. For administrative help and your constant kindness, thank you Diane Stidle.

I am in debt of the company of my other friends at CMU and the CogComp group: Avinava Dubey, Kartik Goyal, Sujay Jauhar, Mrinmaya Sachan, Huiying Li, Dirk Hovy, Jiwei Li, Whitney Houston, Di Wang, Nitish Gupta, Shyam Upadhyay and Stephen Mayhew. Without your levity and friendship, my years in graduate school would have been much colder and more lonesome.

I am particularly grateful to my wife Snigdha for sanity and support, and for holding my feet to the fire in getting my thesis work done. Finally, I am thankful to my parents Archana and Vinod Srivastava, who have supported me in countless ways and provided for an excellent education. All that is the best and worst in me is you.

पिता जी, आपके लिए ...

To my father,
for his love of learning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Learning from language is a fundamental hallmark of human intelligence. People routinely use language to learn new tasks and knowledge. Indeed, verbal and written language forms the core for much of human learning and pedagogy, as reflected in educational devices such as text-books, lectures and student-teacher dialogues. The inextricable role of language in human thought and intelligence has been extensively studied in multiple disciplines including experimental psychology (Parker and Gibson, 1979; Piaget et al., 1959), linguistics (Whorf, 1940; Chomsky, 2006), social anthropology (Dunbar, 1998) and philosophy (Wittgenstein, 1953; Vygotsky, 1980). Language enables strategies including explanations, questions and demonstrations, which facilitate human learning (Figure 1.1 shows an example of such as scenario).

Imagine a child learning to play a game of `tic-tac-toe'. At the onset, a teacher explains the rules of the game to him (e.g., *'At each turn, you can make a cross in an empty cell'*). Once the child learns the rules of the game, the teacher guides him to be a better player by explaining goals or providing advice (e.g., *'If I can make three in a row on my next turn, you should block my move'*). Throughout these experiences, a teacher also instructs the child through demonstrations (e.g., *'If you make this move, you will lose on the next turn'*), as well as answering questions. Eventually, the child becomes an adept tic-tac-toe player. How does he do this? While the child learns some strategies through observing others play and also playing the game himself, the learning process is expedited by the teacher who abstracts essential components of game-playing knowledge, and naturally conveys them through language.

Figure 1.1: Example of human learning via language. The child in the story learns to play a game through a blend of strategies, including explanations, questions and observations

It is then surprising that this type of learning remains underexplored in the fields of artificial intelligence and machine learning. Machine learning systems have so far largely followed a different paradigm of learning: labeling large quantities of data, followed by inductive inference of statistical regularities between output labels and observable representations of the data. While this approach has been extensively successful in a melange of domains and practical applications, there are inherent limitations on what can be learned from labeled examples alone. Consider for example: the number of examples needed by an inductive learner coarsely scales as the log of the size of the hypothesis space (Valiant, 1984). This can be statistically intractable even for common knowledge representations such as ontologies, which children learn with relatively few examples (Gopnik et al., 2004). A central reason for this dissonance is that for the most part, machine learning has ignored richer forms of input, including explanations, clarifications and interaction, which can have significant consequences for learning research. This is the premise that has motivated our work: we believe that to make computer learning as efficient as human learning, we need to develop methods that can learn from natural language interactions.

Further, more than emulating human intelligence or sating scientific curiosity, enabling a paradigm of conversational machine learning can have transformative practical implications. Today's conversational assistants such as Amazon's Alexa[1] or Apple's Siri [2] have the capacity to act on a small number of pre-programmed natural language commands (e.g., "What is the weather going to be like today?"). However, advances in semantic parsing and broader language technologies present the possibility of designing conversational interfaces that enable users to teach computers using language, similar to how humans teach new tasks to one another. For example, if a user wants Alexa to have a new functionality such as *"whenever there is an important email I haven't seen within an hour, read it out to me"*, she should be able to teach this verbally, rather than wait for a software developer to create

[1] `https://en.wikipedia.org/wiki/Amazon_Alexa`
[2] `https://en.wikipedia.org/wiki/Siri`

such a functionality. This instruction may involve explaining what constitutes an *"important email"* for the user, and providing further clarifications and background knowledge. When humans teach other humans, such knowledge is often imparted naturally through explanations, e.g., *"emails from my advisor are usually important"*. If AI assistants could be taught in a similar fashion, this could engender a host of ingenious applications and effectively make every computer user a programmer.

Much of the work described in this dissertation was done as part of a larger effort towards creating such an intelligent personal assistant, which can be programmed using natural language by human users (Labutov et al., 2018). This effort included exploring multiple competencies towards learning from human-like interactions, including learning reusable conditional procedures, learning facts about the world stored in an ontology, and grounding the learned knowledge in perception. In general, multiple types of tasks can be learned through language[3] There can also be other types of information about learning that can be acquired from language, such as meta-knowledge about what to learn, or who to learn from. However, the focus of our work is the specific problem of *learning concepts* (such as the concept of important emails) through natural language.

1.1 Problem statement

This dissertation presents approaches towards enabling computers to learn concepts (or equivalently, binary classification tasks) through natural language interactions, where we define a concept as a Boolean function on some domain of instances. For example, such tasks might include learning the concept of important emails (over the domain of emails), the concept of a malignant tumor (over the domain of tumors), the concept of a 'negative review' (over product reviews), the concept of 'fraud' in credit history analysis, etc. The

[3]'Learning' is a conflated term, which can denote a wide variety of things. For example, it may refer to developing the ability to play a game, memorizing an alphabet or arithmetic tables, performing a procedure such as booking a flight ticket or riding a bike, etc.

ability to automatically learn such concepts is a core cognitive ability, with applications across diverse domains. As we shall see, learning such concepts also finds natural value in scenarios involving computers responding to human instructions. Furthermore, binary classification constitutes the classical setting for supervised machine learning, and hence successful approaches for this problem can be adapted to other supervised learning settings. Our aim in this work is to develop methods in machine learning and language understanding that leverage fundamental aspects of natural language to enable conversational concept learning in artificial agents embodied as sensor-effector systems.

Communicating effectively with computers through natural language has been one of the longstanding goals of AI. The bottleneck in using language to guide machines is the gap between the way modern computers work – by deterministic processing of symbols following an unambiguous and pre-defined set of rules, and the way humans communicate via language – an inherently ambiguous yet powerful information channel. This makes it difficult for machines to interpret human language. Apart from the knowledge of a lexicon of words and their semantics, language interpretation requires resolution of linguistic ambiguities (e.g., issues of polysemy, snyonymy), as well as an understanding of situational context and background knowledge.

However, we believe that advances in NLP and speech research as well as the emergence of ubiquitous computing devices (such as smart-phones and tablets) that can sense the environment enable the possibility of agents that can learn from human language interactions in ways that were not possible till very recently. In this dissertation, by language interpretation we will refer to *semantic parsing* – converting natural language sentences to *logical forms* grounded in a domain specific symbolic language, which represent their meanings, and can be easily understood by machines. For example, a sentence such as "Drop a note to mother saying I'll be late" may be converted to a logical form such as `sendMessage(recipient:mother, text:stringVal(I'll be late))`[4]. While the state-of-

[4]This approach takes a model-theoretic view of semantics, i.e. the space of predicates in the logical

the-art in semantic parsing has made steady advances in recent years, there is a need for NLP methods to expand an awareness of the environment, as well as to drive training of semantic parsers from easily accessible sources of supervision. Thus, conjoined with our primary goal of developing machines that can learn concepts from language, a significant component of this dissertation is the development of new methods for semantic parsing that incorporate non-linguistic context and require weaker forms of supervision.

In this introduction, we discuss conceptual advantages of supplementing machine learning methods with language, enumerate the key challenges involved in learning from language, and outline a conceptual framework for such an agent. We then describe specific components of the framework that work in this dissertation explores.

1.2 Why learn from language?

At the onset, we briefly enumerate conceptual advantages that learning from language can present over conventional methods of learning.

The field of machine learning has traditionally focused on making inductive inferences from static collections of labeled data. However, this type of learning may often seem *unnatural*, especially compared to how humans teach and learn from each other. For example, an administrative assistant might learn how to process travel reimbursement requests through explanation by a supervisor, but would never consider learning this by first collecting volumes of data, then performing inductive analysis to infer the correct procedure.

Secondly, learning from inductive observation is often inadequate since getting large volumes of labeled examples is *infeasible* for the long tail of real-world learning tasks, which may be highly domain-specific, or even personalized (for example, consider the concept of important emails, which can vary from person to person). Collecting personalized labeled

language determines the set of possible meanings it can convey

data for such scenarios might be infeasible.

Another related aspect is that in many cases, inductive learning can be *inefficient* compared to learning from language. Consider learning a concept such as 'emails from my mother', which can be much more easily explained in language than through generalization from labeled examples. For example, one might say 'these emails come from mother@email.com'. In such scenarios language can significantly reduce the sample complexity of learning.

As previously mentioned, perhaps the most significant possibility of agents that can learn interactively from language might be to make AI systems *accessible* to non-experts and everyday people, allowing them to teach personalized new concepts and procedures as needed. The possibility to program computers using natural language coupled with the growing ubiquity of computing devices with formidable repertoires of sensor-effector capabilities can provide fertile ground for a range of creative applications.

1.3 Key Challenges

In this work, our goal is to develop algorithms that can learn through natural language interaction. This presents some key challenges.

1. The first is the problem of **learning to interpret natural language**, which refers to the primary ability to understand language. As mentioned before, in context of this dissertation, this problem is synonymous with semantic parsing of sentences to logical forms. This process involves not only understanding the semantics of words and their composition in natural language, but also aspects of pragmatics of language use in situational contexts (famously embodied in the Gricean maxims (Grice, 1975)), which can affect its meaning. We note here that the ability to understand language need not be perfect, but a basic ability to interpret is a prerequisite for learning from language. To explain, a child might not completely understand all of a teacher's

instructions, but still learn useful behavior. However, a rather unsurprising result from our work is that an improved ability to interpret language generally leads to an improvement in performance on learning tasks.

2. The second challenge is **learning from interpretation**[5], which refers to using interpretations of language to learn and reason about the external environment. These two problems are closely related. The former problem refers to parsing natural language into domain symbols that computers can process, while the latter refers to the processes of going from these domain symbols to the internal generalization/concept definitions (usually in form of a classification model) that the agent uses to represent its learned results. In context of this dissertation, learning from interpretation refers to the different mechanisms through which interpretations of language statements can be used to operationalize learning of binary classification (concept learning) tasks. In different parts of this work, we show that language can drive learning through multiple ways, such as through grounding concepts to features and direct declarative knowledge transfer of concept definitions.

3. While the above two aspects are inextricably linked, a distinct but immensely useful attribute of human language is that it is inherently conversational and interactive. This is especially useful in pedagogical settings, where it allows human learners to dynamically engage with a teacher in activities such as asking questions and seeking clarifications, which can often make learning easier. This also suggests the possibility that agents that can learn from language can be substantially more useful if they can **learn to interact** in mixed-initiative setting, i.e. where the learner can also proactively engage the teacher through dialog, rather than only listen.

The individual components of this dissertation work (briefly described in Section 1.6) address aspects of these problems in the context of concept learning tasks, and hence

[5]Not to be confused with the homonymous but unrelated terminology from Inductive Logic Programming

provide a basic conceptual interface for guiding machine learning algorithms from language.

1.4 Thesis statement

The thesis of this research is that machines can learn concepts grounded in the environment from natural language interactions, in ways similar to humans. In particular, machines can use language to (1) compute rich and expressive features for learning tasks, (2) constrain classification models by leveraging declarative knowledge, and (3) interactively engage with humans to simplify their learning. At the same time, their ability to automatically interpret language can be guided by incorporating different types of linguistic and non-linguistic context. Such context can not only provide discriminative features for semantic parsers, but also reduce the need for expensive labeled training data.

1.5 Framework for Learning from language

In this section, we introduce our framework for learning from language and highlight some specific problems that the work in this dissertation explores. We first explain our view of language and its grounding to sensor-effector agents. Next, we describe our agent architecture for learning from language, and enumerate specific types of knowledge about function approximation tasks that natural language can convey. Finally, through a concrete example, we illustrate sub-problems involved in concept learning from language interaction that we consider in this work.

1.5.1 Grounding language in sensor-effector systems

The grounding of language in the external environment is a fundamentally valuable property that makes it useful above abstract symbol systems (Harnad, 1990). While several prominent approaches such as Distributional Semantics (Firth, 1957) attempt to model

meaning of language in isolation, any approaches towards learning about the environment from language must necessarily ground linguistic symbols to their physical denotations. Consequently, grounding is a pervasive theme throughout our work, which capacitates both the ability to learn about the environment from symbolic representations of language (*learning from interpretation*), as well as the ability to understand language (*learning to interpret*) in situational context of the environment.

In this dissertation, we view an agent as a *sensor-effector* system (Russell et al., 1995). In this view, an agent can interact with the environment by either perceiving the environment through sensors, or acting upon the environment through effectors. Language grounding can be achieved in this framework by simply mapping words and phrases in language to trigger specific sensors or effectors, enabling the agent to observe or modify the environment. To explain, consider a smartphone as a sensor-effector system. It contains a repertoire of sensors that can read emails from an inbox, access stored phone numbers and contact names, note geographical location (GPS), perceive kinematic data (accelerometer), etc. It also contains effectors that can send messages, make phone calls, set alarms, capture an image, etc. A natural language command from the phone user (e.g., *Set an alarm for 5pm today*) can be parsed to a logical form such as `createEvent(type:alarm, time:1700)`. The logical form can thought of a computer program, whose execution is grounded in the external environment (API calls of the alarm-clock application).

Sensors are particularly important from the perspective of concept learning. They provide perceptual inputs that by themselves, or in composition with other sensors, can help realize complex *concept learning*. For example, a geo-positioning sensor can combine with an accelerometer to learn a concept corresponding to 'stuck in traffic'. An important possibility is that new concepts learned once can themselves act as *compound* sensors for further learning. e.g., the concept 'work related emails' could be taught in terms of an existing concept like 'emails from my boss'.

The forementioned framework of sensors and effectors naturally lends itself to realizing

condition-action rules. e.g., 'send my boss an email if I am stuck in traffic'. In principle, enabling people to instruct such rules via language can enable a range of creative applications, essentially converting ordinary human users into programmers. The condition satisfaction in such condition-action rules is essentially a problem of concept identification based on the state of the environment, resolved by observation and categorization of an instance through perceptual sensors. If the condition is true, the required subsequent processing consists of calling the execution of operational procedures that are finally grounded in the effectors.

The process of language interpretation can itself be subsumed in the *sensor-effector* view of intelligent agents by assuming a language understanding/generation unit as a distinct kind of sensor/effector (cognate with a Chomskyian Language Acquistion Device). A language understanding sensor might consist of a semantic parser mappings surface forms of natural language to intrinsic representations (logical forms) that provide basic access to other sensors or effectors. Such ability is essential for the possibility of guiding the agent through natural language. Implicitly, this would consist of a grammar and inference modules that map basic natural language utterances to the agent's internal logical language.

1.5.2 Agent Architecture

Figure 1.2 shows a general framework for learning from language. An agent consists of a language interpreter, a task learner and a persistent Knowledge Base. The environment consists of the ambient surroundings, and also includes a teacher (e.g., a mobile phone user). The agent can receive input from the environment in the form of (1) perceptual observations through sensors, as well as (2) natural language instruction from a teacher. In turn, the agent can also modify the environment through its effectors by performing actions and asking questions. We note that this framing subsumes traditional approaches

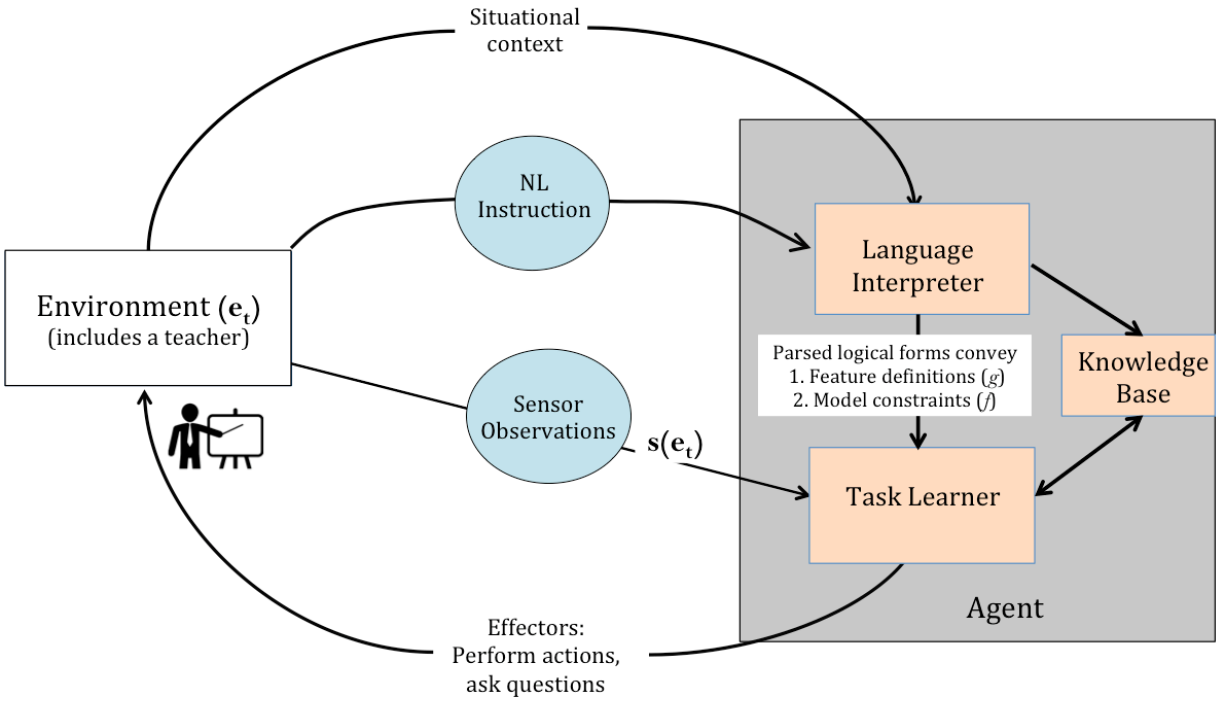


Figure 1.2: Framework for learning from language. An agent consists of a language interpreter, a task learner and a persistent Knowledge Base. The environment consists of the ambient surroundings, and includes a teacher. The agent can perceive the environment through sensors, and can also receive natural language advice from the teacher. The advice can indicate the variables in the environment that are relevant for a learning task (g), or describe the relationship between those variables and the concept to be learned (f)

such as supervised learning (from observations only), active learning (asking a teacher for specific labels), etc.

In general, an agent can also refer to a personal knowledge base of background knowledge to drive its learning, while also modifying the knowledge base with new facts as it continues to learn. While background knowledge is a vital component of human learning and reasoning, we do not delve into this aspect in this work. In this dissertation, learning agents have no persistent memory and only learn a single classification task at a time. This precludes certain types of learning (e.g., curricular learning, multi-task learning, etc.), but allows us to better focus on the task of concept learning from language in isolation.

Formally, an agent can perceive the state of the environment e_t at time t through a set of sensors s. At any time, the agent can access the set of sensor observations $s(e_t)$.

The teacher, who is part of the environment, can provide natural language advice of two types. Firstly, the teacher can describe how to compute the quantities (variables) necessary for learning a target concept in terms of groundable sensor observations, $s(e_t)$. In terms of the learning problem, this corresponds to defining the features of a learning problem, $X = g(s(e_t))$. The learning problem then consists of finding a function approximation $Y = f(g(s(e_t)))$ to predict a concept in the environment, based on sensor readings $s(e_t)$. A second type of natural language advice can directly describe the target function f to be learned. Thus, the scope of knowledge conveyed from language explanations can refer to:

- the sensor observations to use among the set of all possible observations, $s(e)$
- the reasoning or computation required to infer variables of interest (g)
- the mapping from those variables to the target concept (f)

We will illustrate this further through an example in the next section. While we have described concept learning here, the same framework can conceptually also handle other learning tasks framed as function approximation via language, such as regression and reinforcement learning.

At this point, we observe that the different arcs in Figure 1.2 representing the interactions between language, task learning and the environment also broadly correspond to key challenges that we described earlier. The arcs from the environment to the language interpreter correspond to the challenge of *learning to interpret* language in situated contexts in an environment. The arcs from the language interpreter to the task learner correspond to the challenge of *learning from interpretion*, i.e. operationalizing knowledge received through language into task-specific models. Finally, the bottom arc directed from the task learner towards the environment can be seen as signifying the challenge of *learning to interact*.

1.5.3 A Concrete Example

Let us consider an example to concretely understand the problems involved in learning from natural language interactions. Figure 1.3 shows a hypothetical scenario, where an email user explains the concept of important emails to a personal assistant. The interaction exemplifies some common facets of learning from language, including explanations, clarifications, and declarative transfer of knowledge that we explore in this work.

Natural language instruction can encompass a very wide range of useful information, e.g., advice on when to learn, which teachers to trust, what questions to ask during exploratory or confirmatory phases of learning, etc. Thus, it is very hard to have a definitive formulation for learning from language. However, from the perspective of machine learning, language can be especially valuable at three specific points in the learning process, beginning with *identifying relevant features* for learning tasks. Most successful applications of machine learning rely on a process of careful feature engineering, even before any learning takes place. This involves design of features on part of the data analyst, which defines the variables on which the learning problem is based. Instead of writing computer programs to extract such features, learning from natural language instruction can expand the scope of computer systems to learn from people with no machine learning expertise (such as the physician in this example).

Once a learning problem is formulated, language can guide the learning process itself through direct transfer of declarative knowledge about the task to be learned, which may be hard for a machine to learn by itself. Such knowledge can simplify learning and improve generalization by preferring models that concur with human provided advice. In our example, an intelligent learner can use such knowledge (in the form of the email user's explanation) to avoid models that may predict a majority of emails that the user receives to be important. In this sense, language can itself be a source of supervision by *specifying model constraints* through declarative knowledge. This can simplify learning since model

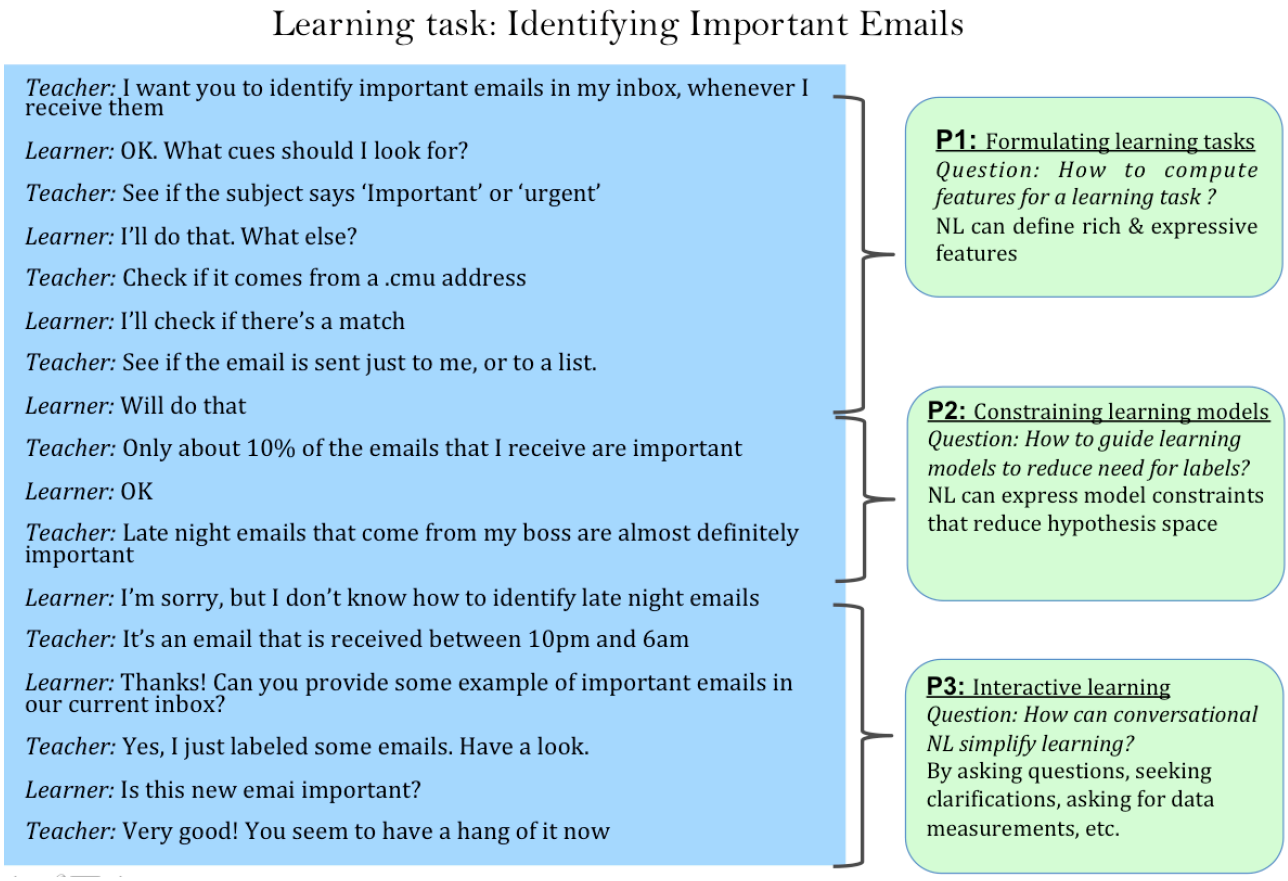


Figure 1.3: Hypothetical scenario of learning from natural language interaction. This examples illustrates many of the problems involved in learning from natural language interactions that we explore in our work. In particular, correct interpreting explanations from a teacher can convey multiple types of useful information about learning tasks: these include *identifying relevant features* for a learning task, as well as directly providing supervision by *specifying model constraints* in the form of declarative knowledge. Apart from simply learning from interpretations of explanations from a teacher, the ability of a learning agent to *interactively seek dialog and ask questions* using language can also accelerate the learning process. At the same time, in order to have these abilities, the agent must also *learn to interpret free-form language in situated contexts*

constraints can guide learning towards more promising parts of the hypothesis space, and ensuring that the correct hypotheses are not spuriously discarded during training.

A third way in which language can facilitate learning is through *interactive dialog* on part of the agent. As seen in the current example, this may takes multiple forms: (1) asking questions about the target concept (2) seeking a clarification of a previous explanation, (3) asking for data measurements or labels to validate a learned model, etc.

At the same time, in order to be able to learn from language, the learner needs the ability to *interpret instructional language* as feature definitions, and identify different types of model constraints and data measurements from free-form language. For example, we assume that a statement such as 'Check if the email is received after 10pm' can be mapped to a logical form such as `greaterThan( getValueForField(timeStamp), 2200)`, which can be executed as a query or a program against an email to retrieve the value of a feature (a boolean value which denotes whether en email was received after 10 pm). As we previously mentioned, a corollary of this grounding is that the scope of meanings that can be conveyed through natural language is determined by the predicates in the domain logical language. Learning to map such instructional language to actionable logical representations is challenging, especially since creating labeled data consisting of language statements paired with annotations of logical forms is expensive.

In this section, we have highlighted some of the principal sub-problems involved in concept learning from language. The research presented in this dissertation develops methods to address some of these. An important observation that we reiterate here is that in many scenarios, learning may depend on reasoning over other accumulated background knowledge, which we do not attempt to model here. For example, the email user might explain that a useful indicator for identifying important emails is whether the user knows the sender of an email. We may not be able to identify this directly, but we may know that this is related to whether the user has previously communicated with the sender (which we likely can observe from previously sent emails), and use that as a proxy feature. Ideally,

an agent would be able to reason and use background knowledge to automatically infer such relevant features. However, for sake of simplicity of analysis, we do not consider the use of such background knowledge in the current work.

1.6 Chapter Overview

In the following chapters we will describe our solutions to the above-mentioned problems. The organization of this dissertation is as follows. **Chapter 2** provides context for the contributions of our research by summarizing previous work on learning from natural language interactions and grounded language acquisition. The bulk of the remaining chapters can be seen as constituting three parts, corresponding to the key challenges described in Section 1.3.

Part I: Learning from Interpretation

The following two chapters summarize our work on *learning from interpretation*.

Chapter 3 introduces an approach for using natural language explanations to compute features for concept learning tasks. The approach depends on the use of a semantic parser to map explanations of concepts from users to logical forms in a domain specific language. The predicted logical forms are evaluated in context of different instances of data, hence operationalizing rich and compositional feature functions. A discriminative classifier is trained on this representation of the data to learn a concept model. The approach is empirically tested on classification tasks to evaluate its effectiveness against baseline methods for text classification.

Chapter 4 presents our research on incorporating declarative knowledge from natural language to supervise training of machine learning models. In particular, we focus on leveraging conditional statements and the semantics of quantifier expressions to learn classifiers in scenarios with limited or no labeled examples of a concept. We present an approach that

converts natural language explanations to quantitative constraints on machine learning models, and describe a procedure that uses posterior regularization to incorporate such constraints into training algorithms for concept classifiers.

Part II: Learning to Interpret

The following two chapters describe our work on new approaches for semantic parsing for *learning to interpret* by incorporating environmental context.

Chapter 5 describes how sensory context can be incorporated to jointly learn both a semantic parser and a concept classifier, extending our approach from Chapter 3. The only supervision consists of a small number of labeled examples of a concept, and our approach extends supervised semantic parsing by learning from a weaker form of supervision than has been previously explored. Our approach suggests that pragmatic context can be a viable and inexpensive source of supervision for training semantic parsers.

Chapter 6 presents an approach for incorporating conversational context to improve language interpretation. Our approach uses a structured prediction formulation that can capture structural regularities in conversation sequences. Our formulations enables training of semantic parsers that incorporates both traditional text-based features that depend on the statement being parser, as well as structural features to model previous conversational history and the flow of discourse in the conversation.

Part III: Learning to Interact

Chapter 7 describes our approach for *learning to interact* in context of concept learning in a conversational setting. Here, we present our research on interactively learning question-asking strategies. We present a reinforcement learning formulation of the problem, where the space of actions consists of different types of questions that a learner can ask a teacher, and reward is evaluated in terms of improvements in the concept model that an asked

question leads to.

Finally, **Chapter 8** summarizes the important contributions of this research, and looks forward to some directions for future work.

Chapter 2

Related Work

In this section, we overview previous approaches in two areas germane to our work. We first discuss prior work in the areas of learning from natural interaction. Next, we look at related work in the field of language interpretation in grounded contexts, focusing on applications where language helps to learn a downstream task.

2.1 Learning from Natural Language

The connection between language and learning has been deeply explored from the perspective of linguistic and cognitive theories (Halliday, 1993; Lemke, 1990). In terms of computer learning, natural language has been explored as a medium for providing instruction starting with early rule-based systems such as SHRLDU (Winograd, 1972). However, these methods were difficult to generalize beyond narrow domains, or handle natural variation in language. More recently, multiple statistical approaches have used different kinds of supervisory signals to use more complex language for performing external tasks in different target environments (Liang et al., 2009b; Branavan et al., 2009). Lau et al. (2009) convert language instructions to user-interface actions on websites. Tellex et al. (2011) learn to map instructions to sequences of actions by a physical robot. There is, in particular, a rich

body of literature on using natural language instructions to help reinforcement learning agents complete tasks efficiently. In many of these approaches, language is used to specify preferences for which actions to choose in specific states (Matuszek et al., 2013; Krening et al., 2017). Similar approaches have also explored parsing natural language instructions from user manuals in game playing frameworks (Branavan et al., 2012; Eisenstein et al., 2009).

Our work is akin in spirit to previous work by Goldwasser and Roth (2014) and Clarke et al. (2010), who train semantic parsers in weakly supervised contexts, where language interpretation is integrated in real-world tasks, such as learning the rules of solitaire. The theme of leveraging the mutually supportive relationship between language and learning about the environment advanced in their research finds reflection in our work. Also notable is the Interactive Task Learning framework (Laird et al., 2017), in which an agent modeled on the SOAR cognitive architecture (Laird, 2012) can learn simple manipulation tasks or extend previously learned tasks through natural language instruction from a human teacher. A distinct body of recent research (Wang et al., 2017, 2016) explores interactive language in simulated block-worlds to incrementally enable grounded language acquisition and instruction for completing manipulation tasks. All of the above mentioned approaches use a semantic parsing component, which maps language instructions to logical forms.

Other approaches have explored the use of language input for tasks such as QA (Sukhbaatar et al., 2015), without explicitly modeling the process of language interpretation as semantic parsing. Similarly, Weston (2016) learn a dialog model from natural language feedback, whereas Ling and Fidler (2017) use natural language feedback to improve training of an image captioning system. More generally, the idea of learning through human interactions has been explored in several settings such as behavioral programming (Harel et al., 2012), natural language programming (Biermann, 1983) and learning by procedures from instruction (Huffman and Laird, 2014; Azaria et al., 2016), etc. Azaria et al. (2016) present a Learning from Interaction Agent (LIA), which can be taught by users to learn new pro-

cedures (operationalized as a sequence of simpler instructions) and generalize these procedures to new scenarios by automatically inducing their arguments. LIA can also learn more complex procedures through composition of already taught procedures.

To the best of our knowledge, ours is the first work to use semantic interpretation to train classifiers from natural language. In terms of providing declarative supervision, this conceptually extends previous approaches such as Generalized Expectation (Mann and McCallum, 2010), Posterior Regularization (Ganchev et al., 2010) and Constraint-driven learning (Chang et al., 2007), which integrate manually provided 'side-information' into training of machine learning models. Our work can also be seen as extending broader paradigms such as Incidental supervision (Roth, 2017) and learning-based programming (Kordjamshidi et al., 2018). However, rather than pre-programmed domain knowledge as in many previous approaches, this information is conveyed to the learner as free-form language.

2.2 Interpreting Grounded Language

In this work, we model the process of language interpretation as semantic parsing, which is a rich area of NLP research. Statistical methods for semantic parsing rely on approaches from structured prediction, and have been explored in diverse domains (Zelle and Mooney, 1996; Miller et al., 1996; Zettlemoyer and Collins, 2005; Artzi and Zettlemoyer, 2013). Semantic parsers have traditionally been trained using datasets consisting of statements paired with labeled logical forms (Zettlemoyer and Collins, 2005). More recent approaches have focused on training semantic parsers from statements paired with denotations of logical forms, rather than logical forms themselves (Liang et al., 2011; Krishnamurthy and Mitchell, 2012; Berant et al., 2013)[1]. Some of our work extends this paradigm by

[1]By denotation, we refer to the non-symbolic outcome of executing a symbolic logical form in a particular context or model theory. e.g., a statement such as 'three less than twenty times six' may be mapped to a logical form such as `minus( prod(20,6), 3)`, which can be executed in context of the domain theory of arithmetic to yield the denotation of the logical form, the integer 117

attempting to learn from still weaker signals, which have not been previously explored.

The semantic parsing algorithms presented in this work aim to incorporate contextual cues from the external environment to improve semantic parsing. The role of such context in assigning meaning to language has previously been emphasized from theoretical perspectives in computational semantics (Bates, 1976; Van Dijk, 1980; Searle, 1969). Subsequently, a substantial body of literature has explored approaches for learning the semantics of language in grounded contexts (Mooney, 2008; Liang et al., 2009b; Fleischman and Roy, 2005; Roy and Reiter, 2005). More recent approaches such as Narasimhan et al. (2015) also use supervision from grounded environments to learn language representations that can help an agent complete tasks in reinforcement learning frameworks.

The methods developed in our work are largely agnostic to the choice of semantic parsing formalism used. However, for this work, our semantic parsers are based on CCG semantic parsing, which is a popular semantic parsing framework (Zettlemoyer and Collins, 2007; Kwiatkowski et al., 2013; Artzi et al., 2015) that maps language to logical commands through function operations in combinatory logic. The CCG formalism (Steedman and Baldridge, 2011) has been widely used for its explicit pairing of syntax with semantics, and allows expression of long range dependencies extending beyond context-free-grammars. However, the same methods could be adapted with work with other popular semantic parsing frameworks, such as Dependency Compositional Semantics (Liang, 2013).

Chapter 3

Interpreting Explanations as Feature Definitions

Our sensory and cognitive faculties enable us to observe features of the environment. Such features play a central role in learning about the environment. The relationship between features and knowledge about the world has a long history. Aristotle (4th Century B.C.) suggested that every idea or concept has a set of necessary features, which specify its essence. Successful applications of modern machine learning too usually rely on a process of feature definition, which transforms raw observations about the data into meaningful inputs that are provided to a learning algorithm (Blum and Langley, 1997).

In this chapter, we suggest that language can be a natural medium for communicating such information, and present an approach for using language to define expressive features for concept learning. For example, learning the concept of 'negative product reviews' can benefit from an explanation such as 'Negative reviews often mention phrases such as "poor quality", "too expensive" and "disappointed"'. One way to interpret such a statement is as defining some boolean property over the space of instances (reviews) X; that is, the statement s defines a predicate $p_s : X \rightarrow \{0, 1\}$. The predicate operationalizes a feature function that will have a non-zero value if the statement applies to a specific instance. e.g.,

if a product review contains the word 'disappointed'. Our work in this chapter provides a simple computational approach to do this.

We posit that the process of mapping from language to such predicates can be accomplished through the use of semantic parsing. Semantic parsing maps language to logical forms, which have previously been explored in an eclectic variety of settings such as question answering (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Berant et al., 2013), robot navigation (Kate et al., 2005) and spreadsheet manipulation (Gulwani and Marron, 2014). Our simple insight here is that logical forms obtained from semantic parsing of concept explanations can be thought of as computer programs that specify the computation of features. They can be evaluated in context of different instances of data, and hence naturally act as feature functions.

There can be significant value in using logical forms to specify and compute features on which machine learning systems can be trained. Perhaps most significantly, instead of hand-crafting such features (as in traditional machine learning), language can provide a natural way to specify the computation of rich and compositional features, reflecting the richness and compositionality of natural language. Suppose we are interested in teaching a computer the concept of work-related emails. Instead of designing features manually, a user might say 'These emails usually come from a CMU address'. By converting such statements to executable programs such as `isStringMatch( currentEmail::getDomainName, stringVal(CMU) )`, we can automatically extract computable features that can characterize a given email. The ability to dynamically define such features for learning tasks can be especially significant from the perspective of enabling non-experts to teach machine learning systems.

In this chapter, we show that such interactive feature space construction can significantly outperform classification models trained on traditional representations of data, notably in scenarios with limited labeled data. We note here that our approach in this chapter assumes that we have labeled data for training semantic parsers, in the form of

statements annotated with their logical forms. In chapter 5, we will extend this approach to provide a method that jointly learns to both parse and classify from a small number of labeled classification examples only.

Work described in this chapter has previously appeared in Srivastava et al. (2017b).

3.1 Related Work

Early work in the field of Artificial Intelligence has explored in detail the role of features for analogical and inductive reasoning. Notable among these explorations are the questions of which among a potentially infinite number of possible features are *relevant* to a concept, and work on the theory of *determinations* (Russell, 1987) that formalizes this. In terms of objective, our approach here aligns with Blum (1990) and Roth and Small (2009), who explore learning of classification tasks through interactive supervision in presence of arbitrarily large feature spaces. Recent work such as by Lake et al. (2015) explores concept learning from few examples in limited settings, and presents encouraging results for one-shot learning by learning representations of instances over Bayesian programs. However, despite rich literature that focuses on feature labeling and feature selection for statistical learning, the issue of feature learning from language (i.e., converting natural language explanations to definitions of feature computation for machine learning methods) remains relatively unexplored.

3.2 Approach

As just mentioned, semantic parsing can be used to map natural language descriptions to logical forms, which can denote feature functions, and can be evaluated in context of different instances to yield the value of a feature. Our premise here is that interpreting natural language explanations from human users as defining relevant and informative features for

classification tasks can improve the sample complexity of concept learning.

Human users are asked to explain a concept using a set of short (sentence-length) natural language explanations. Our approach relies on initially training a semantic parsing model on a dataset of statements paired with logical forms. This semantic parser is subsequently used for predicting interpretations (logical forms) for the user-supplied explanations of concepts. These logical forms are evaluated on labeled examples of the concept learning task, which provides a new feature representation of these examples. A discriminative classifier is then trained over this new data representation to learn a classification model for the concept in terms of the user-specified features. Figure 3.1 illustrates how a learned parsing model can used to interpret explanations as specifying feature functions, and be used to learn concept classifiers from labeled data.

System Input and Output: The preliminary input to the system consists of a semantic parsing grammar $\mathcal{G}$ and a set of n_{train} statements paired with their logical forms $\{(s_j, l_j)\}^{n_{train}}$ to train a semantic parser. The initial output is a semantic parsing model (specified by parameters θ_p), which can be used to parse explanations.

Once the parsing model is trained, the system can be used to learn concept models. Learning a classifier for a new concept requires a set of n natural language statements describing the concept and a small number of labeled examples of the concept, $\{(x_i, y_i)\}^m$. The output is a concept classifier model (specified by parameters θ_c).

The remainder of this section describes these steps in detail.

3.2.1 Training a Semantic Parser

We train a semantic parser to map an explanation sentence s like '*The subject contains the word postdoc*' to a logical form l like `getPhraseMention(subject, stringVal(`postdoc'))`. Logical forms can be evaluated in a context of an instance x (here, an email) to yield some

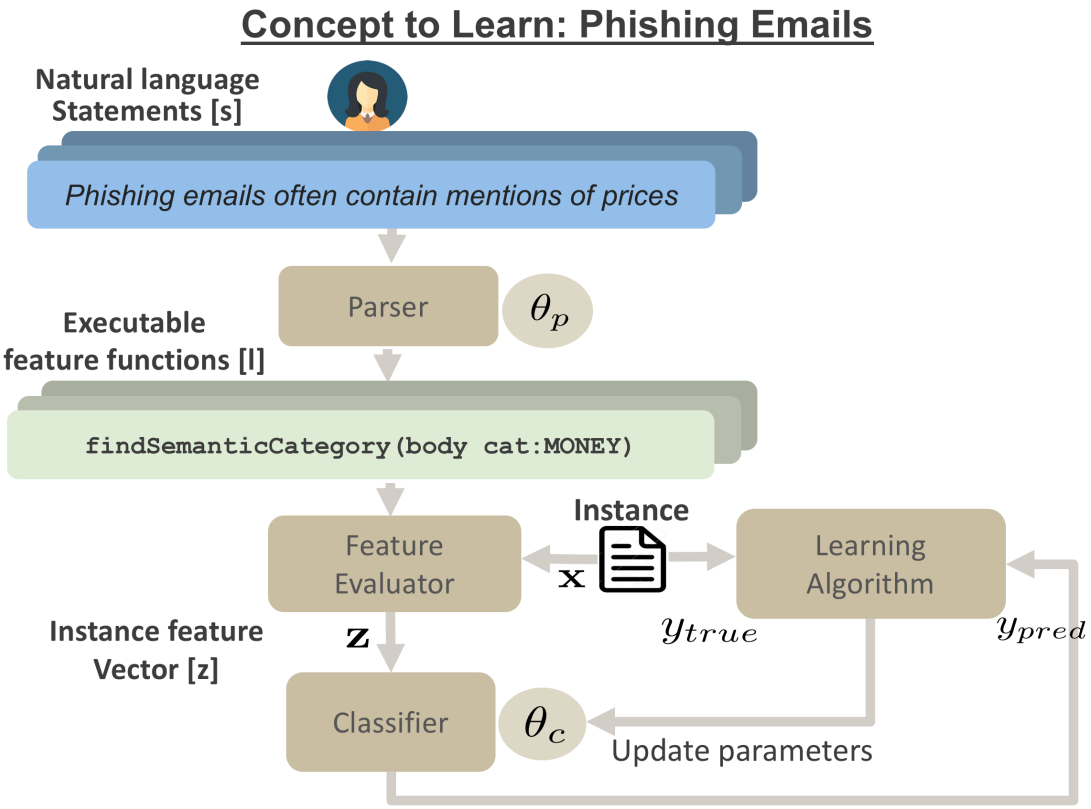


Figure 3.1: Natural language explanations can express rich and expressive feature definitions for classification tasks. A user describes a concept such as 'phishing emails' using language explanation. These are parsed by a semantic parser to logical forms, which are in turn evaluated in the context of new instances to yield a feature vector. A discriminative classifier (with parameters θ_c) is then trained on this representation of the data using a small number of labeled examples for the concept learning task

meaningful output $[\![l]\!]_x$ (whether the statement is true for an email). The predicates (such as `stringVal`) and constants (such as `subject`) come from a pre-specified logical language.

Logical Language: Our testbed in this work is concepts about emails. Hence, we specify a logical language that is expressive enough to be useful for concept learning in this domain. Table 3.1 lists the predicates in our logical language along with descriptions of their evaluation, and some illustrative examples showing how they can represent the meaning of natural statements. We include a special predicate (`unknown`) to label statements whose meanings go beyond our logical language (last row in Table 3.1), essentially ignoring them. Such statements compose about 25% of our data. Note that this logical language can express compositional meanings. e.g., '*These inquiries will usually seek a postdoc opportunity and include a CV*' can be expressed as `and (getPhrasesLike(email, stringVal(`seek postdoc opportunity')), (stringMatch attachment (stringVal`CV')))`. The evaluations of some predicates uses NLP tools that go beyond exact keyword matching. In Section 3.4, we show that it is language understanding (semantic parsing), rather than these resources, which enables learning from natural explanations.

Grammar: The grammar, $\mathcal{G}$, for a semantic parser produces the set of possible candidate logical forms ($\mathcal{G}(s)$) for any natural language sentence s. Semantic parsing grammars are usually specified by a lexicon containing mappings from words to symbols in the logical language, and a set of combinatory syntactic rules. Different semantic parsing formalisms vary in the degree of permissivity of these lexical and syntactic rules. For this work, we use CCG based semantic parsing, a popular semantic parsing approach Zettlemoyer and Collins (2005); Artzi et al. (2015) that tightly couples syntax with semantics. For the CCG grammar, we manually compile a small domain lexicon containing a list of trigger words mapped to their syntactic categories and associated logical predicates. e.g. {'subject', NP, `subject`}. We then use the PAL lexicon induction algorithm (Krishnamurthy, 2016) to expand the lexicon and induce syntactic combination rules by adding automatically

Predicate	Description and evaluation
`stringVal`	Returns string value corresponding to a text span in the statement
`getPhraseMention`	Looks for matching tokens or phrases in a text, and return true if an exact match is found. e.g., *The subject contains the word postdoc* → `getPhraseMention(subject,stringVal('postdoc'))`
`getPhrasesLike`	Uses an alignment based Textual Entailment model to find the closest semantic match for a phrase in a text. Uses distributional semantics to identify semantically similar words. Returns true if a match is found. e.g., *The emails often want me to buy something* → `getPhrasesLike(email,stringVal(`buy something'))`
`getSemanticCategory`	Looks for occurrences of pre-specified semantic categories in a target text (identified with Stanford CoreNLP's expanded NER tagger), and returns true if a match is found. e.g., *these emails often have contain prices and quotes* → `getSemanticCategory(body, MONEY)`
`stringMatch`	Returns true if one string value contains another. e.g., *Spam emails are rarely from a yahoo or gmail address* → `not( or(stringMatch(sender, stringVal(`yahoo')), stringMatch(sender, stringVal(`gmail'))))`
`stringEquals`	Returns true if two string values are equal
`or/ and/ not`	Boolean predicates with usual interpretations
`beginWith/endWith`	Return true if a target text contains a phrase, a similar phrase, or a semantic category at its beginning/end. e.g., *The emails often mention a phone number at the end* → `endWith(body, NUMBER)`
`merge`	Combines multiple elements into a list. e.g., *These emails often refer to problems like baldness and aging.* → `getPhrasesLike(email, merge(stringVal(`baldness'), stringVal(`aging')))`
`before`	Returns true if there is an instance of one type preceding an instance of another type in a text
`length`	Lengths of lists or text fields (in number of words)
$\geq$, `equals`	Usual arithmetical comparators
`unknown`	Return false by default. Used to deal with statements that cannot be reasonably expressed using predicates in the language. e.g., *These emails are from weird addresses.* → `unknown`

Table 3.1: Predicates in logical language used by our semantic parser for learning of email based concepts

generated entries.

Model Training: Semantic parsing can be seen as a structured prediction problem of predicting the highest scoring logical form l for a given natural language sentence s. Following traditional semantic parsing models(Zettlemoyer and Collins, 2005), we are interested in linear predictors of form:

$$\hat{l} = \underset{l \in \mathcal{G}(s)}{\text{argmax}} \ \theta_p^T \psi(s, l)$$

Here, model training corresponds to learning a real valued parameter vector θ_p for this structured prediction problem. $\psi(s, l)$ denotes a real valued vector of semantic parsing features, which can depend on the sentence to be parsed (s) and a candidate logical form (l). The data for training the parser is a collection of n_{train} tuples $\{(s_j, l_j)\}^{n_{train}}$ of statements paired with logical forms. For training the parser, we use the Structured Perceptron algorithm (Collins, 2002). We follow the feature set from Zettlemoyer and Collins (2007), consisting of (i) indicator features for lexicon entries and (ii) indicator features for rule applications. We also include (iii) string based features denoting the number of words in a string span, and whether a string spans occur at the beginning or end of the utterance. For retrieving the best parses for a statement, we use beam search with a beam size of 500.

3.2.2 Training a Concept Classifier

The semantic parsing model trained in the previous section is used to predict the logical form l for an explanation s of a concept. A set of n such explanations yields n corresponding logical forms. These logical forms are used in conjunctions with a set of m labeled instances of the concept to learn a classifier for the concept. In particular, each logical form is evaluated in context of an instance of the data to yield the value of a feature $z = [\![l]\!]_x$. This leads to a an new n dimensional feature representation of the instances $Z_{m \times n}$, which can be used to train a classifier from a set of labeled examples of the concept, $\{(x_i, y_i)\}^m$.

For the discriminative classifier, we choose a logistic regression model. The classifier receives as input a feature representation previously described, and returns a parameter vector θ_c for the learned logistic regression model, which can be used to predict whether an new instance belongs to a concept.

3.3 Data

Our evaluation for this task focused on the domain of emails as a testbed for concept learning from language. This choice was motivated by our belief that concept learning from explanations can be particularly effective for email classification; this is a domain where (1) target concepts are often highly individualized, and (2) labeled data can often be scarce.

We created a dataset of 1,030 emails paired with 235 natural language statements made by human users in the process of teaching a set of seven concepts. The dataset was collected using the Amazon Mechanical Turk crowd-sourcing platform. We deployed two tasks:

1. A *Generation* task requiring workers to create original emails
2. A *Teaching* task requiring workers to write statements that characterize a concept

We chose to create a new email corpus for evaluation of our approach, rather than use an existing corpus such as Enron (For et al., 2004), since we wanted diverse examples representative of everyday concepts that most people would be able to understand as well as teach to a computer. Much of the Enron corpus is highly specific and contextualized, making it difficult to teach for an outsider.

In this section, we describe the data and the two tasks in more detail.

Concept	# of emails	Prompt
CONTACT	167	"You are writing an email to yourself to personally keep note of a person contact"
EMPLOYEE	149	"You are a boss writing an email to your employee requesting something to be done"
EVENT	138	"You are writing an email to a friend asking to meet up at some event"
HUMOR	134	"You are writing an email to a friend that includes something humorous from the Internet"
MEETING	142	"You are writing an email to a colleague trying to request a meeting about something"
POLICY	146	"You are writing an office email regarding announcement of some new policy"
REMINDER	154	"You are writing an email to yourself as a reminder to do something"

Table 3.2: Email concepts used in our experiment, together with the prompts used to describe the concept to workers. The same prompt was used in both the (i) *Generation task* for generating emails belonging to each concept and the (ii) *Teaching task* for collecting natural language statements that explain the concepts

3.3.1 Generation task

This task consisted of a web-page resembling a traditional email composition form (with fields: *recipient*, *subject*, *body*, *attachment*), requiring workers to compose emails in a grounded setting. For this task, we recruited 146 workers residing in the United States. The workers were presented with each of the seven concepts in a sequence, where each concept was represented by a short prompt encouraging workers to imagine a scenario (e.g., a boss writing a request to an employee) and write a hypothetical email.

Table 3.2 shows details of email concepts and the corresponding prompts shown to turkers for the Generation task. Workers were instructed to be realistic (e.g., they were asked to include an attachment if an email is likely to have an attachment in reality), but also creative (to encourage diversity) in composing their emails.

Figure 3.2 shows some samples of emails created by turkers from the Generation task.

Subject: Please do today
From: mike@initech-corp.com
To: melissa@initech-corp.com
Body: Melissa, I noticed you still haven't logged in the injections you did during the flu clinic last Friday. Please do this ASAP. Billing is waiting on this to process the insurance payments. Thanks, Mike Manager, INITECH CORPORATION
Attachment: none

Subject: New airline policy
From: mary@initech-corp.com
To: all-staff@initech-corp.com
Body: Dear all, Please be advised that we are required to use Air France for all flights between the US and France starting January 1. A copy of the announcement is attached. No exceptions. - Mary
Attachment: airlinepolicy122016.pdf

Subject: get together on hiring coordination
From: mary@initech-corp.com
To: virginia@initech-corp.com
Body: Virginia – Need to meet with you about coordinating hiring decisions between our two teams so we avoid skill duplication and get the best new hires. I'm out of the office Friday through Monday. Let me know when we can meet. Suggest a half hour. - Mary
Attachment: none

Subject: Beerfest!
From: mary@initech-corp.com
To: janedoe@gmail.com
Body: Hey, J! I got two tickets for Beerfest next week at the Convention Center - would you want to meet up and go with me? I know we had a blast year and I saw the tickets on sale, so I figured you might want to join. If you can make it let me know! Maybe we can even grab dinner after. Mary
Attachment: none

Figure 3.2: Examples of emails generated by turkers in the Generation task. Turkers were presented with short prompts describing email categories. They were then asked to imagine a scenario and write a hypothetical email belonging to the email category in a web-page resembling a traditional email composition form

3.3.2 Teaching task

The teaching task was then deployed to collect natural language statements that people use to teach a concept to a machine. Workers were presented with five randomly selected concepts using the same prompts (Table 3.2) used in the *Generation* task.

Protocol for Collection of Explanations

For each email concept, a small sample of 20 emails were shown in a style resembling a traditional email inbox (Figure 3.3) to illustrate the concept. Half of the emails were positive examples of the concept to be described (these emails were highlighted and "starred"), and the remaining half were sampled randomly from the other concepts. Workers were encouraged to peruse through the emails while creating up to five statements explaining the concept. Prior to the task, the workers were also informed that the statements are to be used to teach email concepts to a computer. A follow-up quiz assessed an understanding of the task, and contributions from workers with low scores were filtered. The final data contains between 30 and 35 statements describing each category. The data described here is available at `http://www.cs.cmu.edu/~shashans/resources/emails`.

We continue to follow the same data collection methodology for all experiments described later in this dissertation – namely consisting of first showing Mechanical Turkers examples and non-examples of a concept, and asking them to describe these concepts in their own words. A notable observation about this methodology is that turkers have access to examples (and non-examples) of a concept while they are describing the concept. An alternative design choice might ask turkers to explain a concept *after* they have seen examples, but when they don't have access to examples at hand. It is possible that the exact choice of the teaching framework may affect the quality and nature of explanations. However, we do not explore this question in this work.

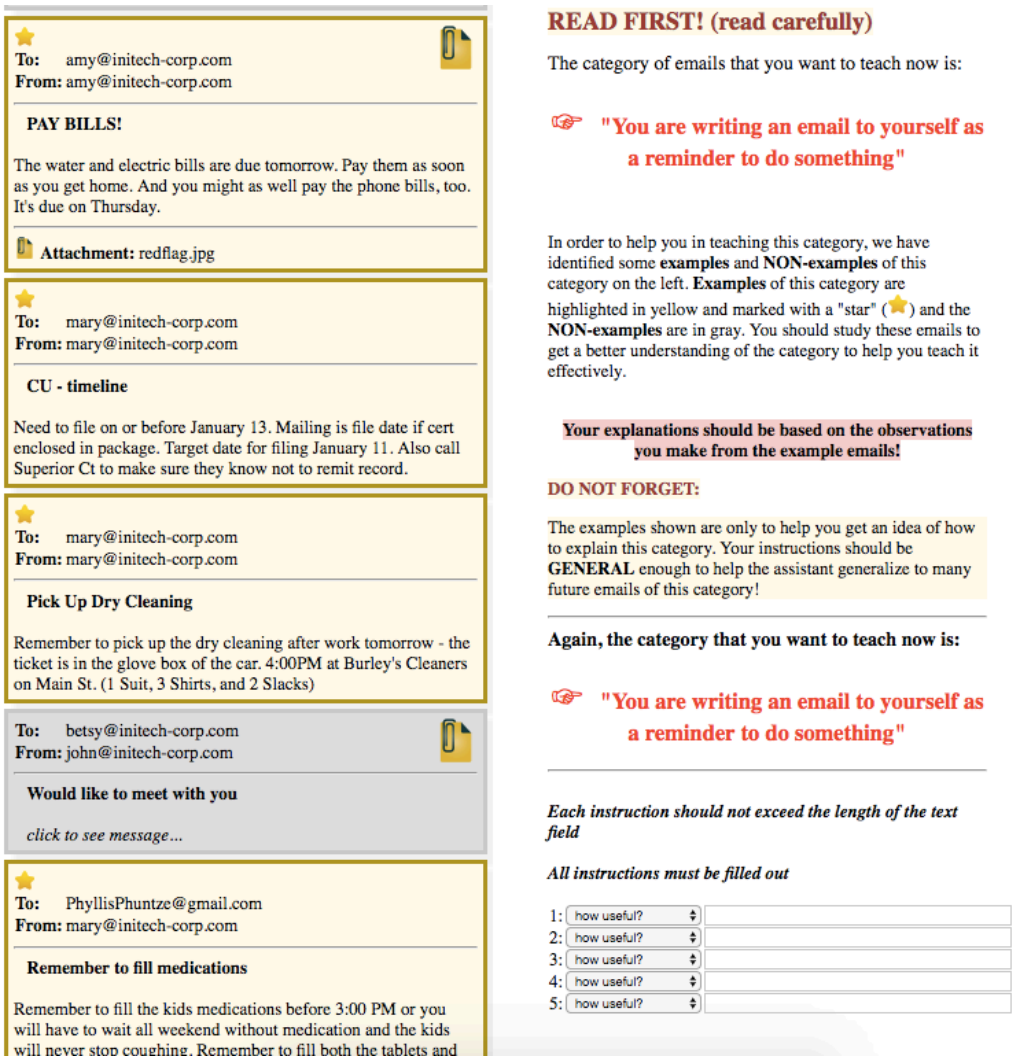


Figure 3.3: Screenshot of *Teaching* task used to collect explanations describing concepts. Each worker is given a concept prompt, together with a set of emails (starred emails denote positive examples of the concept). A turker can enter upto five statements characterizing the concept

These emails usually closes with a name or title
Some reminders will have a date and time in the subject
The body of the email may say funny, picture, or internet
Messages to friends sometimes have jpg attachments
Emails from a public domain are not office requests

Table 3.3: Examples of explanations collected from turkers during the Teaching task

3.4 Evaluation

In this section, we describe baselines and evaluate the performance of our approach for concept learning.

3.4.1 Baselines

Our baselines include the following two classes of approaches:

Text-only models: These approaches classify emails solely based on their text content, and do not have access to natural language explanations of the concept.

- *BoW:* A logistic regression (LR) classifier trained over a bag-of-words representation of emails.
- *BoW tf-idf:* A logistic regression classifier trained over a bag-of-words representation of emails, with tf-idf weighting.
- *Para2Vec:* A logistic regression classifier trained over a distributed representation of email text, learned using deep neural network approach by Le and Mikolov (2014).
- *Bigram:* A logistic regression classifier trained over a representation that also incorporates bigram features. Such as baseline is known to be competitive on several text classification tasks (Wang and Manning, 2012).
- *ESA:* A logistic regression classifier trained over ESA (Explicit Semantic Analysis) representations of emails (Gabrilovich and Markovitch, 2007), which describe a text in terms of Wikipedia topics.

Models incorporating Statements: These approaches also take into account the natural language explanations, in addition to the text of the emails.

- *RTE:* This uses a Textual Entailment model that computes a score for aligning of each statement to the text of each email. A logistic regression is trained over this

representation of the data.

- *Keyword filtering:* Filters based on keywords are common in email systems. We add this as a baseline by manually filtering statements referring to occurrences of specific keywords. Such statements compose nearly 30% of the data. We train a logistic regression over this representation.

We note here that the baseline approaches and our approach all use different types of supervision and resources, and hence a direct comparison is not completely fair. In particular, we note that the text-only models only use traditional supervision (in the form of concept labels) for model training. Our approach, which we refer to as **LNL** (for **Learning from Natural Language**) also presumes the availability of natural language explanations of concepts, as well as a trained semantic parsing model. In this sense, LNL requires strictly stronger supervision than the text-only approaches. However, we note that the training of the parser is a one-time need only, and a parsing model once trained can be used for learning of a large number of possible concepts in a domain (such as email). Further, in Chapter 5 we show that the learning of such a parser can be accomplished from very weak sources of supervision.

Other baselines that also incorporate statements too rely on other types of supervision and resources. In particular, the RTE model relies heavily on lexical semantic representations of words that require large text corpora for learning. The Keyword filtering approach also needs minimal semantic parsing.

3.4.2 Concept-learning Performance

Table 3.4 shows classification performance of our approaches for Learning from Natural Language (*LNL*) against baselines described above for $n = 10$ labeled examples. The reported numbers are average F1 scores over 10 data draws. We observe that *Bigram* and bag-of-word methods are the most competitive among the baselines. On the other hand,

	Contact	Employee	Event	Humor	Meeting	Policy	Reminder	Average
BoW	0.510	0.354	0.381	0.484	0.455	0.588	0.415	0.455
BoW tf-Idf	0.431	0.379	0.402	0.513	0.392	0.576	0.399	0.441
Para2Vec	0.238	0.191	0.121	0.252	0.222	0.286	0.092	0.200
Bigrams	0.525	0.385	0.426	0.525	0.458	0.668	0.423	0.487
ESA	0.187	0.209	0.107	0.194	0.154	0.160	0.131	0.164
RTE	0.551	0.353	0.406	0.475	0.398	0.522	0.232	0.419
Keyword filtering	0.521	0.429	0.412	0.425	0.702	0.748	0.392	0.522
LNL	0.648*	0.381	0.627*	0.565*	0.770*	0.892*	0.470	0.622
LNL-Gold	0.661	0.397	0.677	0.572	0.777	0.917	0.487	0.641
LNL+BoW	0.640	0.421	0.622	0.755	0.731	0.909	0.554	0.661
LNL-Gold+BoW	0.667	0.449	0.659	0.798	0.771	0.927	0.595	0.695

Table 3.4: Concept learning performance (F1 scores) using $n = 10$ labeled examples. Reported numbers are averages over 10 data draws. Columns indicate different concept learning tasks defined over emails. * for the rows corresponding to LNL denotes statistical significance over the best performing non-LNL model

Para2Vec doesn't perform well, probably due to the relatively small scale of the available training data, while *ESA* fails due to the lack of topical associations in concepts.

However, most significantly, we observe that *LNL* dramatically outperforms all baselines for nearly all concepts (except Employee), and shows a 30% relative improvement in average F1 over other methods ($p < 0.05$, Paired Permutation test). Our dataset also contains manual annotations of the explanation statements with logical forms. In Table 3.4, *LNL-Gold* denotes the classification performance with using these annotated gold logical forms (rather than from a trained parsing model). This corresponds to the hypothetical case where the classifier knows the correct semantic interpretation of each natural language sentence from an oracle. These provide a further 2% improvement in concept learning performance (Section 3.4.4 further explores the relationship between parsing and classification performance). We also observe that *LNL* models perform significantly better than *Keyword filtering* ($p < 0.05$), indicating that the model leverages the expressiveness of our logical language.

Finally, the last two rows in the table show performance when the *LNL* methods also utilize BoW representations of the data. The further gains over the base *LNL* models

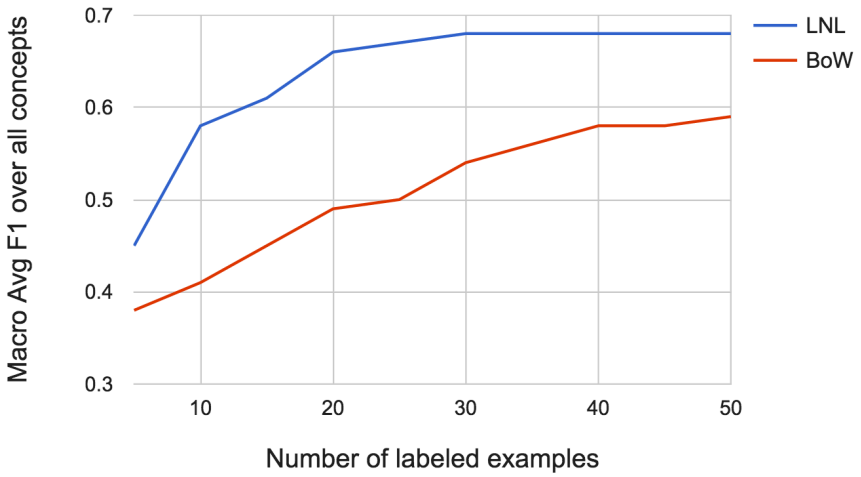


Figure 3.4: Figure showing Avg F1 accuracy over all concepts vs Number of labeled training examples

suggest that original feature representations and natural language explanations contain complementary information for many concepts. These results demonstrate that interpreting natural language statements as feature functions leads to significant improvements in concept learning, and validate our method.

3.4.3 Effect of training set size

A significant motivation for this work is the promise of natural language explanations in facilitating concept learning with a relatively small number of examples. Figure 3.4 shows the dependence of concept learning performance of *LNL* on the number of labeled training examples (size of training set). We observe that while our approach consistently outperforms the bag-of-words model (*BoW*), *LNL* also requires fewer examples to reach near optimal performance, before it plateaus. In particular, the generalization performance for *LNL* is more robust than *BoW* for $n < 10$.

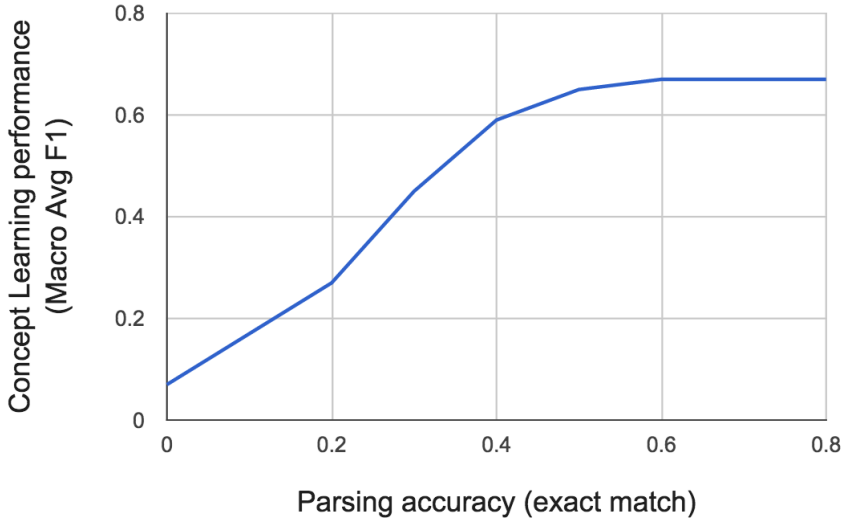


Figure 3.5: Figure showing concept classification performance vs parsing accuracy. This denotes a basic point: better language interpretation leads to better concept learning performance

3.4.4 Concept learning vs language interpretation

To delineate the relationship between parsing performance and concept learning more clearly, we plot concept classification performance for different levels of semantic parsing proficiency in Figure 3.5. For this, we choose the gold annotation logical form for a statement with a probability corresponding to the semantic parsing accuracy, or randomly select a candidate logical form with a uniform probability otherwise for all the statements in our data.

The figure shows an expectedly strong association between parsing performance and concept learning, although gains from parsing taper after a certain level of proficiency. This is partially explained by the fact that natural statements in our data often contain overlapping information, and that the set of statements in our data set may not be sufficient to achieve perfect classification accuracy.

3.5 Discussion

In this chapter, we demonstrated how interactive language can be used to specify relevant features for concept learning tasks. Through experiments, we showed that this strategy can lead to faster learning than traditional classification methods for email classification.

An important aspect of such explanations is that they can not only highlight specific attributes of the data relevant for a learning task (which is more related to the problem of *feature selection*), but also compositionally define new features, which can lead to representations of the data that make inductive learning easier. Thus, for example, a person explaining the concept of an email that is a reminder to oneself might say *'The sender and the recipient address are the same'*. The statement not only also highlights a relevant property from a potentially enormous space of possibilities; but the structure of language often also naturally conveys the structure of computation required to calculate those features.

An important note about the language interpretation approach presented in this chapter is that a new logical language would have to be created (and a corresponding semantic parsing model trained) for each new domain. Designing the predicates for a domain language (and creating a corresponding lexicon) might require significant effort. However, this would be alleviated by the fact that this would only be a one-time effort, which would find re-use across the long tail of possible concept learning problems in a domain.

Chapter 4

Incorporating User Advice as Probabilistic Model Constraints

Traditional supervised learning requires large quantities of labeled examples for generalization. While this paradigm has been widely successful in a large range of applications, this is problematic since obtaining labeled data for user-specific learning tasks can often be infeasible. This is especially relevant from the perspective of enabling ubiquitous machine learning, allowing users to teach personalized concepts (e.g., identifying 'important emails' or 'project-related emails') in scenarios with limited or no training data.

On the other hand, language is often rich in declarative knowledge, which can directly provide supervision for machine learning models. In particular, humans often teach and learn using instructional language in the form of conditional rules expressing relationships between features and labels (e.g., 'If the subject of an email threatens to close an account, it is definitely spam'). By identifying the types of relationships such explanations express, and quantitatively using such information in computational frameworks, we can minimize the labeled examples needed. We posit that such knowledge can enable learning in scenarios with limited or even no labeled examples.

In this chapter, we will focus on leveraging conditional rules and quantifier expressions

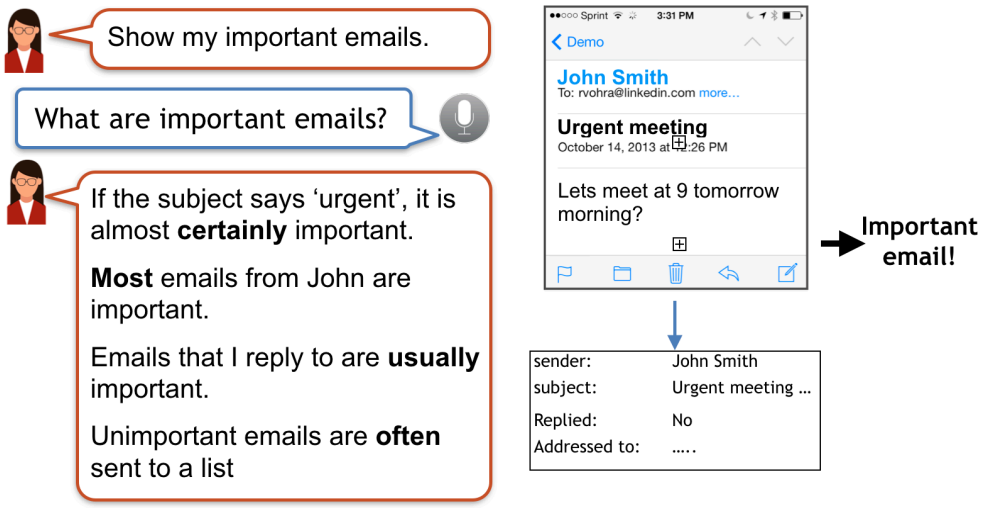


Figure 4.1: Declarative supervision from language can enable concept learning from limited or even no labeled examples. Our approach assumes the learner has sensors that can extract attributes from data, such as those listed in the table, and language that can refer to these sensors and their values

in language to provide supervision for concept learning tasks. Everyday language is rich in quantifiers (such as determiners like 'all', 'each', 'few', and frequency adverbs like 'always', 'usually', 'never'), which are explicit denoters of generality. Since learning is largely synonymous with generalization, it is natural to use such signals to expedite learning. Even as traditional logic has studied the simplest of such quantifiers ($\forall$ and $\exists$) in significant detail, statistical models have not yet leveraged their predictive potential.

We present a very general quantitative framework through which a set of explanations of a concept can be used to learn a classifier from declarative language. For illustration, consider the hypothetical example of a user explaining the concept of an "important email" through natural language statements (Figure 4.1). Our framework takes a set of such natural language explanations describing a concept (e.g., "emails that I reply to are usually important") and a set of unlabeled instances as input, and produces a binary classifier (for important emails) as output. Our hypothesis is that natural language explanations of concepts encode key properties that can aid statistical learning. These include specification of relevant attributes (e.g., whether an email was replied to), relationships between

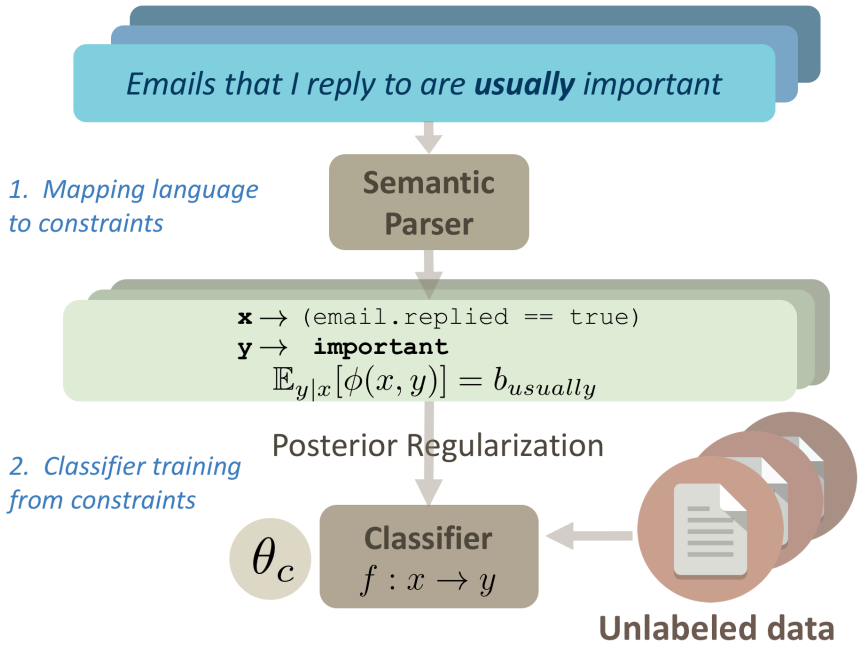


Figure 4.2: Our approach to Zero-shot learning from Language. Natural language explanations on how to classify concept examples are parsed into formal constraints relating features to concept labels. The constraints are combined with unlabeled data, using posterior regularization to yield a classifier

such attributes and concept labels (e.g., if a reply implies the class label of that email is 'important'), as well as the strength of these relationships (e.g., via quantifiers like 'often', 'sometimes', 'rarely'). We infer these properties automatically, and use the semantics of linguistic quantifiers to drive the training of classifiers *without labeled examples for any concept*[1]. This is a novel scenario, where previous approaches in semi-supervised and constraint-based learning are not directly applicable. This is because these approaches require manual pre-specification of expert knowledge for model training. In our approach, this knowledge is automatically inferred from noisy natural language explanations from a user.

The schematic in Figure 4.2 summarizes our methodology. First, we map the set of natural language explanations of a concept to logical forms that identify the attributes

[1]Note that we do not aim to provide a semantic representation to model the issue of quantification in linguistics. Rather, the focus is on leveraging the generalization potential of quantifiers to enable learning from language

mentioned in the explanation, and describe the information conveyed about the attribute and the concept label as a quantitative constraint. This mapping is done through semantic parsing. The logical forms denote quantitative constraints, which are probabilistic assertions about observable attributes of the data and unobserved concept labels. For sake of simplicity, we assume that the strength of a constraint is assumed to be specified by a linguistic quantifier (such as 'all', 'some', 'few', etc., which reflect degrees of generality of propositions). Next, we train a classification model that can assimilate these constraints by adapting the posterior regularization framework (Ganchev et al., 2010).

Intuitively, this can be seen as defining an optimization problem, where the objective is to find parameter estimates for the classifier that do not simply fit the data, but also agree with the human provided natural language advice to the greatest extent possible. Since logical forms can be grounded in a variety of sensors and external resources, an explicit model of semantic interpretation conceptually allows the framework to subsume a flexible range of grounding behaviors.

Work described in this chapter has previously appeared in Srivastava et al. (2018) and Srivastava et al. (2017c).

4.1 Related Work

Many notable approaches have explored incorporation of background knowledge into the training of learning algorithms. However, none of them addresses the issue of learning from natural language. Prominent among these are the Constraint-driven learning (Chang et al., 2007), Generalized Expectation (Mann and McCallum, 2010) and Posterior Regularization (Ganchev et al., 2010) and Bayesian Measurements (Liang et al., 2009a) frameworks. All of these require domain knowledge to be manually programmed in before learning. Similarly, Probabilistic Soft Logic (Kimmig et al., 2012) allows users to specify rules in a logical language that can be used for reasoning over graphical models. More recently, multiple

approaches have explored few-shot learning from perspective of term or attribute-based transfer (Lampert et al., 2014), or learning representations of instances as probabilistic programs (Lake et al., 2015).

Other work (Lei Ba et al., 2015; Elhoseiny et al., 2013) considers language terms such as colors and textures that can be directly grounded in visual meaning in images. Some previous work (Srivastava et al., 2017b) has explored using language explanations for feature space construction in concept learning tasks, where the problem of learning to interpret language, and learning classifiers is treated jointly. However, this approach assumes availability of labeled data for learning classifiers. Also notable is recent work by Andreas et al. (2017), who propose using language descriptions as parameters to model structure in learning tasks in multiple settings. More generally, learning from language has also been previously explored in tasks such as playing games (Branavan et al., 2012), robot navigation (Karamcheti et al., 2017), etc.

Natural language quantification has been studied from multiple perspectives in formal logic (Barwise and Cooper, 1981), linguistics (Löbner, 1987; Bach et al., 2013) and cognitive psychology (Kurtzman and MacDonald, 1993). While quantification has traditionally been defined in set-theoretic terms in linguistic theories[2], our approach joins alternative perspectives that represent quantifiers probabilistically (Moxey and Sanford, 1993; Yildirim et al., 2013). To the best of our knowledge, this is the first work to leverage the semantics of quantifiers to guide statistical learning models.

4.2 Approach

Our approach relies on first mapping natural language descriptions to quantitative constraints that specify statistical relationships between observable attributes of instances and their latent concept labels (Step 1 in Figure 4.2). These quantitative constraints are then

[2]e.g., *'some A are B'* $\Leftrightarrow A \cap B \neq \emptyset$

imbued into the training of a classifier by guiding predictions from the learned models to concur with them (Step 2). We use semantic parsing to interpret sentences as quantitative constraints, and adapt the posterior regularization principle for our setting to estimate the classifier parameters. Next, we describe these steps in detail. Since learning in this work is largely driven by the semantics of linguistic quantifiers, we call our approach **Learning from Natural Quantification**, or **LNQ**.

4.2.1 Mapping language to constraints

A key challenge in learning from language is converting free-form language to representations that can be reasoned over, and grounded in data. For example, a description such as '*emails that I reply to are usually important*' may be converted to a mathematical assertion such as $P(important \mid replied : true) = 0.7$', which statistical methods can reason with. Here, we argue that this process can be automated for a large number of real-world descriptions. In interpreting statements describing concepts, we infer the following key elements:

1. *Feature* x, which is grounded in observed attributes of the data. For our example, 'emails replied to' can refer to a predicate such as `replied:true`, which can be evaluated in context of emails to indicate the whether an email was replied to.

   Incorporating compositional representations enables more complex reasoning. e.g., '*the subject of course-related emails usually mentions CS100*' can map to a composite predicate like '`isStringMatch(field:subject, stringVal(`CS100'))`' , which can be evaluated for different emails to reflect whether their subject mentions 'CS100'. In Chapter 3, we have explored how mapping language to such executable feature functions can be effective for concept learning. As before, here we assume that a statement refers to a single feature, but the method can be extended to handle more complex descriptions.

2. *Concept label* y, specifying the class of instances a statement refers to. For binary classes, this reduces to examples or non-examples of a concept. For our running example, y corresponds to the positive class of important emails.

3. *Constraint-type* asserted by the statement. We argue that most concept descriptions belong to one of three categories shown in Table 4.2, and these constitute our vocabulary of constraint types for this work. For our running example ('emails that I reply to are usually important'), the type corresponds to $P(y \mid x)$, since the syntax of the statement indicates an assertion conditioned on the feature indicating whether an email was replied to. On the other hand, an assertion such as 'I usually reply to important emails' indicates an assertion conditioned on the set important emails, and therefore corresponds to the type $P(x \mid y)$.

4. *Strength* of the constraint. We assume this to be specified by a quantifier. For our running example, this corresponds to the adverb 'usually'. In this work, by *quantifier* we specifically refer to frequency adverbs ('usually','rarely', etc.) and frequency determiners ('few', 'all', etc.).[3] Our thesis is that the semantics of quantifiers can be leveraged to make statistical assertions about relationships involving attributes and concept labels. One way to do this might be to simply associate point estimates of probability values, suggesting the fraction of truth values for assertions described with these quantifiers. Table 4.1 shows probability values we assign to some common frequency quantifiers for English. These values were set simply based on the authors' intuition about their semantics, and do not reflect any empirical distributions. See Figure 4.9 for empirical distributions corresponding to some linguistic quantifiers in our data. While these probability values maybe inaccurate, and the semantics of these quantifiers may also change based on context and the speaker, they can still serve as a strong signal for learning classifiers since they are not used as hard

[3]This is a significantly restricted definition, and does not address non-frequency determiners (e.g.,'the', 'only', etc.) or mass quantifiers (e.g. 'a lot', 'little'), among other categories.

Frequency quantifier	Probability
all, always, certainly, definitely	0.95
usually, normally, generally, likely, typically	0.70
most, majority	0.60
often, half	0.50
many	0.40
sometimes, frequently, some	0.30
few, occasionally	0.20
rarely, seldom	0.10
never	0.05

Table 4.1: Probability values we assign to common linguistic quantifiers (hyper-parameters for method)

Type	Example description	Conversion to Expectation Constraint
$P(y \mid x)$	Emails that I reply to are usually important	$\mathbb{E}[\mathbb{I}_{y=important,reply(x):true}] - p_{usually} \times \mathbb{E}[\mathbb{I}_{reply(x):true}] = 0$
$P(x \mid y)$	I often reply to important emails	$\mathbb{E}[\mathbb{I}_{y=important,reply(x):true}] - p_{often} \times \mathbb{E}[\mathbb{I}_{y=important}] = 0$
$P(y)$	I rarely get important emails	Same as $P(y\|x_0)$, where x_0 is a constant feature

Table 4.2: Common constraint-types, and their representation as expectations over feature values

constraints, but serve to bias classifiers towards better generalization.

We use a semantic parsing model to map statements to formal semantic representations that specify these aspects. For example, the statement 'Emails that I reply to are usually important' is mapped to a logical form like `(x→replied:true y→positive type:y|x quant:usually)`.

Semantic Parser components

Given a descriptive statement s, the parsing problem consists of predicting a logical form l that best represents its meaning. In turn, we formulate the probability of the logical form l as decomposing into three component factors: (i) probability of observing a feature and concept labels l_{xy} based on the text of the sentence, (ii) probability of the type of the assertion l_{type} based on the identified feature, concept label and syntactic properties of the

sentence s, and (iii) identifying the linguistic quantifier, l_{quant}, in the sentence.

$$P(l \mid s) = P(l_{xy} \mid s)\ P(l_{type} \mid l_{xy}, s)\ P(l_{quant} \mid s)$$

We model each of the three components as follows: by using a traditional semantic parser for the first component, training a Max-Ent classifier for the constraint-type for the second component, and looking for an explicit string match to identify the quantifier for the third component.

Identifying features and concept labels, l_{xy}

For identifying the feature and concept label mentioned in a sentence, we presume a linear score $\mathcal{S}(s, l_{xy}) = {\theta_{p_1}}^T \psi(s, l_{xy})$ indicating the goodness of assigning a partial logical form, l_{xy}, to a sentence s. Here, $\psi(s, l_{xy}) \in \mathbb{R}^n$ are semantic parsing features that can depend on both the sentence and the partial logical form, and $\theta_{p_1} \in \mathbb{R}^n$ is a parameter weight-vector for this component of the semantic parser. Following recent work in semantic parsing (Liang et al., 2011), we assume a loglinear distribution over interpretations of a sentence.

$$P(l_{xy} \mid s) \propto {\theta_{p_1}}^T \psi(s, l_{xy})$$

Provided data consisting of statements labeled with logical forms, the model can be trained via maximum likelihood estimation, and be used to predict interpretations for new statements. For training this component, we use a CCG semantic parsing formalism, and follow the feature-set from Zettlemoyer and Collins (2007), consisting of simple indicator features for occurrences of keywords and lexicon entries. This is also compatible with the semantic parsing formalism in Chapter 3, whose data (and accompanying lexicon) are also used in our evaluation. For other datasets with predefined features, this component is learned easily from simple lexicons consisting of trigger words for features and labels. We also identify whether a feature x is negated, through the existence of a `neg` dependency relation with the head of its text-span. e.g., *Important emails are usually not deleted.*

This component is the only part of the parser that is domain-specific. We note that while this component assumes a domain-specific lexicon (and possibly statement annotated with logical forms), this effort is one-time-only, and will find re-use across the possibly large number of concepts in the domain (e.g., email categories).

Identifying assertion type, l_{type}

The principal novelty in our semantic parsing model is in identifying the type of constraint asserted by a statement. For this, we train a MaxEnt classifier, which uses positional and syntactic features based on the text-spans corresponding to feature and concept mentions to predict the constraint type. We extract the following features from a statement:

1. Boolean value indicating whether the text-span corresponding to the feature x precedes the text span for the concept label y.
2. Boolean value indicating if sentence is in passive (rather than active) voice, as identified by the occurrence of `nsubjpass` dependency relation.
3. Boolean value indicating whether head of the text-span for x is a noun, or a verb.
4. Features indicating the occurrence of conditional tokens ('if', 'then' and 'that') preceding or following text-spans for x and y.
5. Features indicating presence of a linguistic quantifier in a `det` or an `advmod` relation with syntactic head of x or y.

Since the constraint type is determined by syntactic and dependency parse features, this component does not need to be retrained for new domains. In this work, we trained this classifier based on a manually annotated set of 80 sentences describing classes in the small UCI Zoo dataset Lichman (2013), and used this model for all experiments.

Identifying quantifiers, l_{quant}:

Multiple linguistic quantifiers in a sentence are rare, and we simply look for the first occurrence of a linguistic quantifier in a sentence, i.e. $P(l_{quant}|s)$ is a deterministic function. We note that many real world descriptions of concepts lack an explicit quantifier. e.g., '*Emails from my boss are important*'. In this work, we ignore such statements for the purpose of training. Another treatment might be to models these statements as reflecting a default quantifier, but we do not explore this direction here.

Finally, the decoupling of quantification from logical representation is a key design decision in the approach. At the cost of linguistic coarseness, this allows modeling quantification irrespective of the logical representation (lambda calculus, predicate-argument structures, etc.).

4.2.2 Classifier training from constraints

In the previous section, we described how individual explanations can be mapped to probabilistic assertions about observable attributes (e.g., the statement 'Emails that I reply to are usually important' may map to $P(y = important \mid replied = true) = p_{usually}$). Here, we describe how a set of such assertions can be used in conjunction with unlabeled data to train classification models.

Our approach relies on having predictions from the classifier on a set of unlabeled examples ($X = \{x_1 \dots x_n\}$) agree with human-provided advice (in form of constraints). The unobserved concept labels ($Y = \{y_1 \dots y_n\}$) for the unlabeled data constitute *latent variables* for our method. The training procedure can be seen as iteratively inferring the latent concept labels for unlabeled examples so as to agree with the human advice, and updating the classification models by taking these labels as given. While there are multiple approaches for training statistical models with constraints on latent variables, here we use the Posterior Regularization (PR) framework. The PR objective can be used to optimize

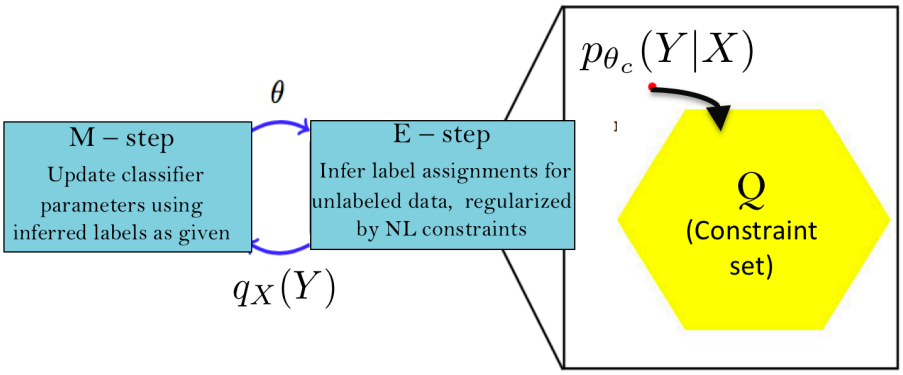


Figure 4.3: We invoke posterior regularization to incorporate quantitative constraints in the training of concept learning models. The procedure can be seen as a modified EM algorithm, where the latent variables correspond to concept label assignments for a set of unlabeled examples. Each natural language explanation is mapped to a quantitative constraint (in form of a probability assertion about features and label values). The conjunction of such constraints define the set Q. In the modified E-step, we prefers concept label assignments that do not violate the constraints (Figure adapted from Ganchev et al. (2010))

a latent variable model subject to a set of constraints, which specify preferences for values of the posterior distributions $p_{\theta_c}(Y \mid X)$.

$$J_Q(\theta_c) = \mathcal{L}(\theta_c) - \min_{q \in Q} KL(q \mid p_{\theta_c}(Y|X)) \tag{4.1}$$

Here, the set Q represents a set of *preferred* posterior distributions over latent variables Y, and is defined as $Q := \{q_X(Y) : \mathbb{E}_q[\phi(X, Y)] \leq \mathbf{b}\}$. The overall objective consists of two components, representing how well does a concept model θ_c explain the data (log-likelihood term $\mathcal{L}(\theta_c)$), and how far it is from the set Q (KL-divergence term).

In our case, each parsed statement defines a probabilistic constraint. The conjunction of all such constraints defines Q (representing models that exactly agree with human-provided advice). Thus, optimizing the objective reflects a tension between choosing models that increase data likelihood, and emulating language advice.

Conversion to PR constraints

The set of constraints that PR can handle can be characterized as bounds on expected values of functions (ϕ) of X and Y (or equivalently, from linearity of expectation, as linear inequalities over expected values of functions of X and Y). To use the framework, we need to ensure that each constraint type in our vocabulary can be expressed in such a form.

Following the plan in Table 4.2, each constraint type can be converted in an equivalent form $\mathbb{E}_q[\phi(X, Y)] = b$, compatible with PR. In particular, each of these constraint types in our vocabulary can be expressed as equations about expectation values of joint indicator functions of label assignments to instances and their attributes. To explain, consider the assertion $P(y = \mathit{important} \mid \mathit{replied} : \mathit{true}) = p_{\mathit{usually}}$. The probability on the LHS can be expressed as the empirical fraction $\frac{\sum_i \mathbb{E}[\mathbb{I}_{y_i=\mathit{important},\mathit{replied}:\mathit{true}}]}{\sum_i \mathbb{E}[\mathbb{I}_{\mathit{replied}:\mathit{true}}]}$, which leads to the linear constraints seen in Table 4.2 (expected values in the table hide summations over instances for brevity). Here, $\mathbb{I}$ denote indicator functions. Thus, we can incorporate probability constraints into our adaptation of the PR scheme.

Learning and Inference

We choose a loglinear parameterization for the concept classifier.

$$p_{\theta_c}(y_i \mid x_i) \propto \exp(y{\theta_c}^T x) \tag{4.2}$$

The training of the classifier follows the modified EM procedure described in Ganchev et al. (2010). As proposed in the original work, we solve a relaxed version of the optimization that allows slack variables, and modifies the PR objective with a L_2 regularizer. This allows solutions even when the problem is over-constrained, and the set Q is empty (e.g.

due to contradictory advice).

$$J'(\theta_c, q) = \mathcal{L}(\theta_c) - KL(q|p_{\theta_c}(Y|X)) \\ - \lambda \, ||\mathbb{E}_q[\phi(X,Y)] - b||^2 \tag{4.3}$$

The key step in the training is the computation of the posterior regularizer in the E-step.

$$\underset{q}{\text{argmin}} \; KL(q \mid p_{\theta_c}) + \lambda \, ||\mathbb{E}_q[\phi(X,Y)] - b||^2 \tag{4.4}$$

This objective is strictly convex, and all constraints are linear in q. We follow the optimization procedure from Bellare et al. (2009), whereby the minimization problem in the E-step can be efficiently solved through gradient steps in the dual space. In the M-step, we update the model parameters for the classifier based on label distributions q estimated in the E-step. This simply reduces to estimating the model parameters θ_c for the logistic regression classifier, when class label probabilities are known. In all experiments, we run EM for 20 iterations and use a regularization coefficient of $\lambda = 0.1$.

4.3 Data

For evaluating our approach for learning classifiers from declarative language, we created datasets of classification tasks paired with descriptions of the classes, as well as used some existing resources. In this section, we summarize these steps.

4.3.1 Shape classification

To experiment with our approach in a wider range of controlled settings, part of our evaluation focuses on synthetic concepts. For this, we created a set of 50 shape classification tasks that exhibit a range of difficulty, and elicited language descriptions spanning

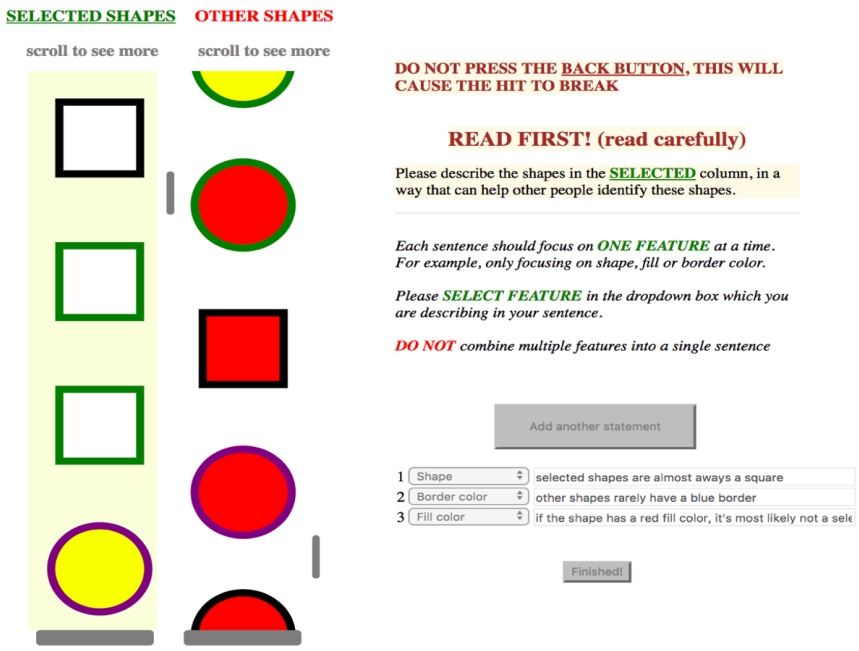
a variety of quantifier expressions. The tasks require classifying geometric shapes with a set of predefined attributes (fill color, border, color, shape, size) into two concept-labels (abstractly named 'selected shape', and 'other'). The datasets were created through a generative process, where features x_i are conditionally independent given the concept-label. Each feature's conditional distribution is sampled from a symmetric Dirichlet distribution, and varying the concentration parameter α allows tuning the noise level of the generated datasets (quantified via their Bayes Optimal accuracy[4]). A dataset is then generated by sampling from these conditional distributions. We sample a total of 50 such datasets, consisting of 100 training and 100 test examples each, where each example is a shape and its assigned label.

For each dataset, we then collected statements from Mechanical Turk workers that describe the concept. The task required turkers to study a sample of shapes presented on the screen for each of the two concept-labels (see Figure 4.4(a)). They were then asked to write a set of statements that would help others classify these shapes without seeing the data. In total, 30 workers participated in this task, generating a mean of 4.3 statements per dataset.

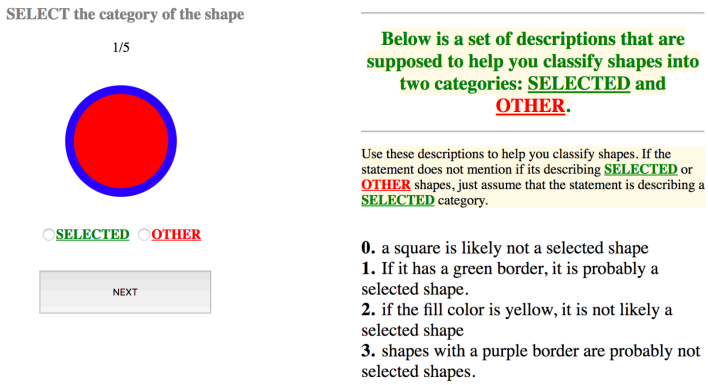
4.3.2 Email categorization

In Chapter 3, we describe a dataset of language explanations from human users describing 7 categories of emails, as well as 1030 examples of emails belonging to those categories. While that work uses labeled examples, and focuses on mapping natural language explanations ($\sim$30 explanations per email category) to compositional feature functions, we can also use statements in the same data for evaluating our approach. Even while language quantifiers were not studied in the previous work, we found about a third of the statements in this data to mention a quantifier.

[4]This is the accuracy of a theoretically optimal classifier, which knows the true distribution of the data and labels



(a) Statement generation task



(b) Quiz for human evaluation

Figure 4.4: Shapes data: Mechanical Turk tasks for (a) collecting concept descriptions, and (b) human evaluation from concept descriptions

Shapes: If a shape doesn't have a blue border, it is probably not a selected shape. Selected shapes occasionally have a yellow fill. Most of the other shapes are triangles.
Emails: Emails that mention the word 'meet' in the subject are usually meeting requests Personal reminders almost always have the same recipient and sender Policy announcements typically contain a pdf attachment
Birds: A specimen that has a striped crown is likely to be a selected bird. Birds in the other category rarely ever have dagger-shaped beaks Most of the selected birds have a solid tail pattern

Table 4.3: Examples of explanations for each domain

4.3.3 Bird species identification

The CUB-200 dataset (Wah et al., 2011) contains images of birds annotated with observable attributes such as size, primary color, wing-patterns, etc. We selected a subset of the data consisting of 10 species of birds and 53 attributes (60 examples per species). Turkers were shown examples of birds from a species, and negative examples consisting of a mix of birds from other species, and were asked to describe the classes (similar to the Shapes data). During the task, users also had access to a table enumerating groundable attributes they could refer to (see Figure 4.5). In all, 60 workers participated, generating 6.1 statements per task on average.

The datasets for Bird species identification and Shape classification described here are available at `http://www.cs.cmu.edu/~shashans/resources/lnq`.

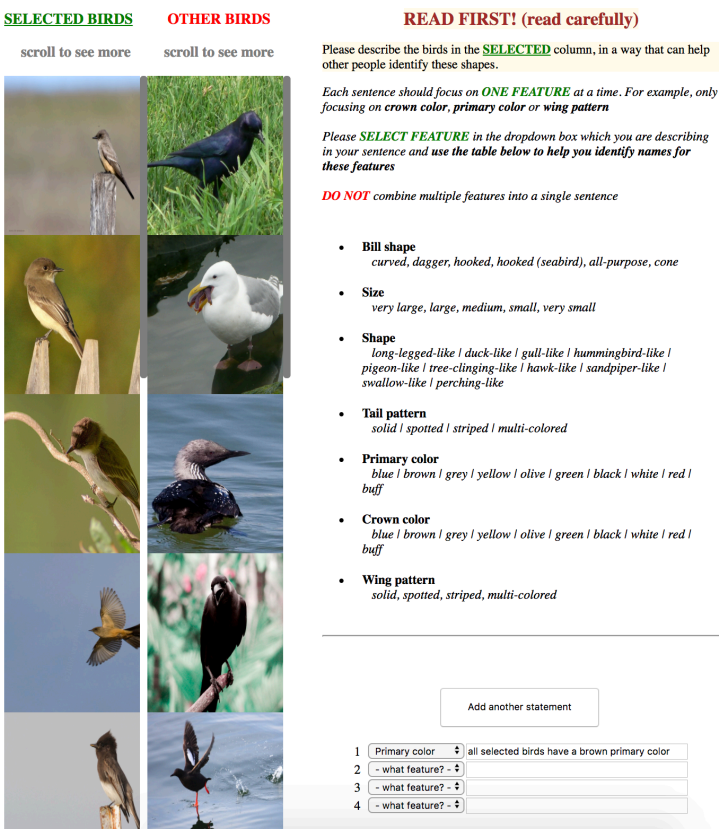


Figure 4.5: Statement generation task for Birds data

4.4 Evaluation

Incorporating constraints from language has not been addressed before, and hence previous approaches for learning from limited data such as Mann and McCallum (2010); Chang et al. (2007) would not directly work for this setting. Our baselines hence consist of extended versions of previous approaches that incorporate output from the parser, as well as fully supervised classifiers trained from a small number of labeled examples.

4.4.1 Classification performance

The top section in Table 4.4 summarizes performance of various classifiers on the Shape datasets, averaged over all 50 classification tasks. FLGE+ refers to a baseline that uses the Feature Labeling through Generalized Expectation criterion, following the approach

Approach	Avg Accuracy	Labels	Descriptions
LNQ	0.751	no	yes
Bayes Optimal	0.831	–	–
FLGE+	0.659	no	yes
FLGE	0.598	no	yes
LR	0.737	yes	no
Random	0.524	–	–
Ablation:			
LNQ (coarse quant)	0.679	no	yes
LNQ (no quant)	0.545	no	yes
Human:			
Human teacher	0.802	yes	writes
Human learner	0.734	no	yes

Table 4.4: Classification performance for various learning strategies on Shapes datasets (averaged over 50 classification tasks).

in Druck et al. (2008); Mann and McCallum (2010). The approach is based on labeling features are indicating specific class-labels, which corresponds to specifiying constraints of type $P(y|x)$[5]. While the original approach Druck et al. (2008) sets this value to 0.9, we provide the method the quantitative probabilities used by LNQ. Since the original method cannot handle language descriptions, we also provide the approach the concept label y and feature x as identified by the parser. FLGE represents the version that is not provided quantifier probabilities. LR refers to a supervised logistic regression model trained on $n = 8$ randomly chosen labeled instances.[6] We note that LNQ performs substantially better than both FLGE+ and LR on average. This validates our modeling principle for learning classifiers from explanations alone, and also suggests value in our PR-based formulation, which can handle multiple constraint types. We further note that not using quantifier probabilities significantly deteriorates FLGE's performance.

Figure 4.6 provides a more detailed characterization of LNQ's performance. Each blue

[5]In general, Generalized Expectation can also handle broader constraint types, similar to Posterior Regularization

[6]LNQ models are indistinct from LR w.r.t. parameterization, but trained to maximize a different objective. The choice of n here is arbitrary, but is roughly twice the number of explanations for each task in this domain

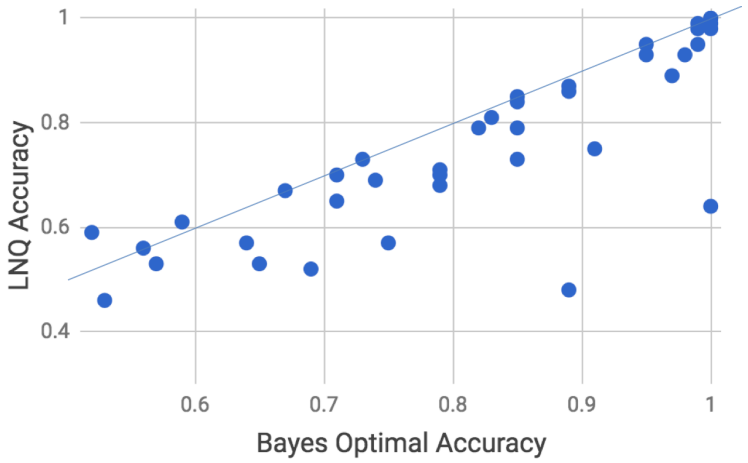


Figure 4.6: LNQ vs Bayes Optimal Classifier performance for Shape learning tasks. Each dot represents performance of the approach on a dataset generated from a known distribution. Dots towards the right correspond to progressively easier learning tasks (as quantified by the Bayes Optimal accuracy). The diagonal line represents the trajectory of an asymptotically optimal (Bayes Optimal) Classifier. Dots that lie above the line correspond to a learned classifier performing better than the Bayes Optimal on a dataset (a sample size effect).

dot represents performance on a shape classification task. The horizontal axis represents the accuracy of the Bayes Optimal classifier, and the vertical represents accuracy of the LNQ approach. The blue line represents the trajectory for $x = y$, representing an ideal statistical classifier in the asymptotic case of infinite samples. We note that LNQ is effective in learning competent classifiers for all levels of hardness. Secondly, except for a small number of outliers, the approach works especially well for learning easy concepts (towards the right). From an error-analysis, we found that a majority of these errors are due to problems in parsing (e.g., missed negation, incorrect constraint type) or due to poor explanations from the teacher (bad grammar, or simply incorrect information).

Figure 4.7 shows results for email classification tasks. In the figure, LN* refers to the approach in Chapter 3, which uses natural language descriptions to define compositional features for email classification, but does not incorporate supervision from quantification. For this task, we found very few of the natural language descriptions to contain quantifiers for some of the individual email categories, making a direct comparison impractical. Thus

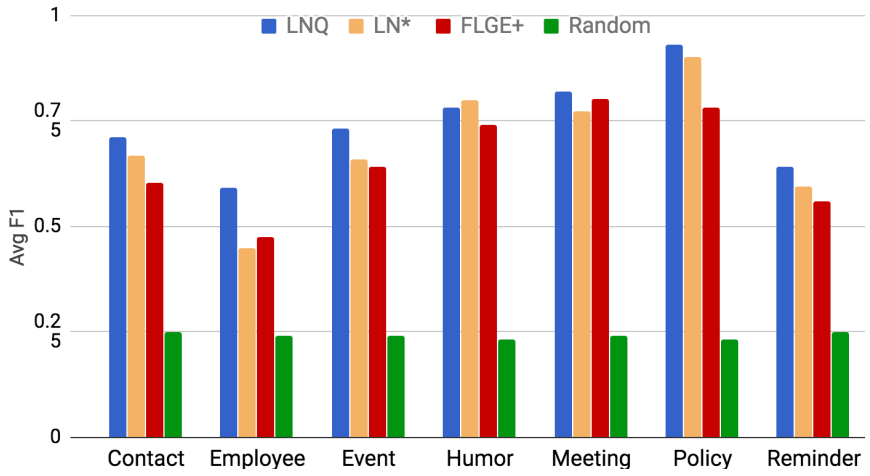


Figure 4.7: Classification performance (F1) on Email Categorization tasks.

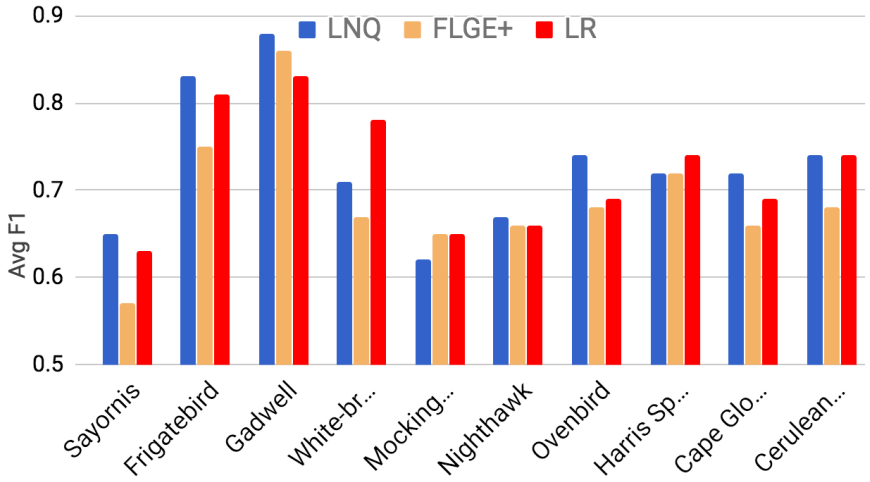


Figure 4.8: Classification performance (F1) on Birds Species Identification tasks.

in this case, we evaluate methods by combining supervision from descriptions in addition to 10 labeled examples (also in line with evaluation in the original paper). We note that additionally incorporating quantification (LNQ) consistently improves classification performance across email categories. On this task, LNQ improves upon FLGE+ and LN* for 6 of the 7 email categories.

Figure 4.8 shows classification results on the Birds data. Here, LR refers to a logistic regression model trained on n=10 examples. The trends in this case are similar, where LNQ consistently outperforms FLGE+, and is competitive with LR.

4.4.2 Ablating quantification

From Table 4.4, we further observe that the differential associative strengths of linguistic quantifiers are crucial for our method's classification performance. LNQ (no quant) refers to a variant that assigns the same probability value (average of values in Table 4.1), irrespective of quantifier. This yields a near random performance, which is what we'd expect if the learning is being driven by the differential strengths of quantifiers. LNQ (coarse quant) refers to a variant that rounds assigned quantifier probabilities in Table 4.1 to 0 or 1. (i.e., quantifiers such are *rarely* get mapped to 0, while *always* gets mapped to a probability of 1). While its performance (0.679) suggests that simple binary feedback is a substantial signal, the difference from the full model indicates value in using soft probabilities. On the other hand, in a sensitivity study, we found the performance of the approach to be robust to small changes in the probability values of quantifiers.

4.4.3 Comparison with human performance

For the Shapes data, we evaluated human teachers' own understanding of concepts they teach by evaluating them on a quiz based on predicting labels for examples from the test set (see Figure 4.4(b)). Second, we solicit additional workers that were not exposed to examples from the dataset, and present them only with the statements describing that data (created by a teacher), which is comparable supervision to what LNQ receives.

We then evaluate their performance at the same task. From Table 4.4, we note that a human teacher's average performance is significantly worse ($p < 0.05$, Wilcoxon signed-rank test) than the Bayes Optimal classifier indicating that the teacher's own synthesis of concepts is noisy. The human learner performance is expectedly lower, but interestingly is also significantly worse than LNQ. While this might be potentially be caused by factors such as user fatigue, this might also suggest that automated methods might be better at reasoning with constraints than humans in certain scenarios. However, we note that

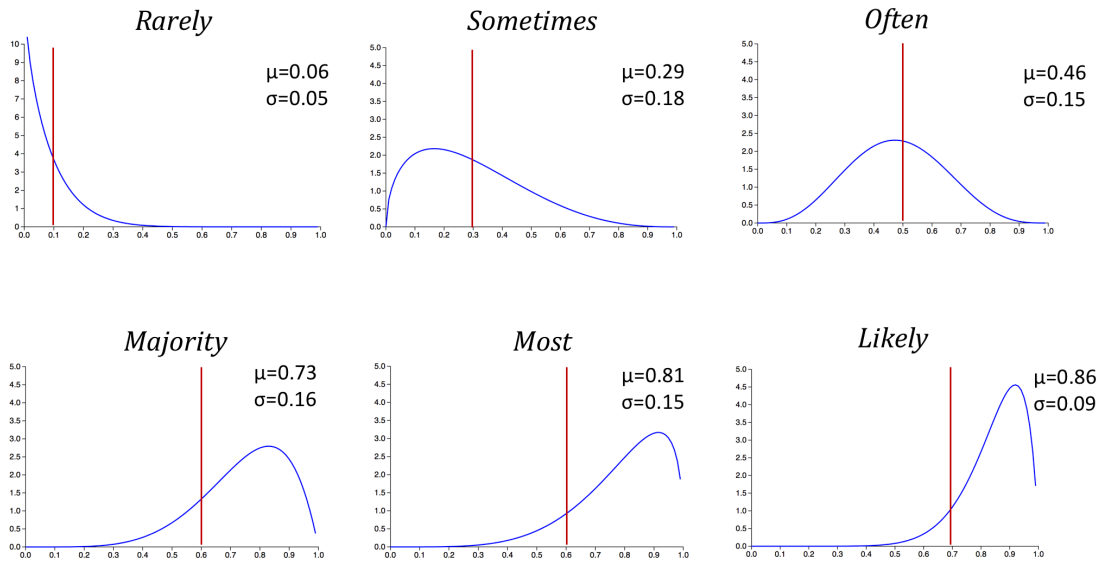


Figure 4.9: Empirical distributions of probability values corresponding to statements mentioning six quantifiers (on Shapes data). Plots show Beta distributions with Method-of-Moment estimates. Vertical bars correspond to pre-registered estimates of point probability values from Table 4.1. The differences between the pre-registered beliefs and empirical probabilities suggest that LNL can be effective for learning classification tasks even when the empirical semantics of quantifiers are relatively inaccurately modeled, or are too diffused to be meaningfully represented by point probability estimates

the Shapes data present a significantly simplified learning task, where there are only a small number of pre-identified features, and background knowledge is not useful due to the abstract nature of the concepts to be learned. In this sense, these results need to be validated through comprehensive experiments in more domains and a broader range of learning tasks .

4.4.4 Empirical semantics of quantifiers

We can estimate the distributions of probability values for different quantifiers from our labeled data. For this, we aggregate sentences mentioning a quantifier, and calculate the empirical value of the (conditional) probability associated with the statement, leading to a set of probability values for each quantifier. Figure 4.9 shows empirical distributions of probability values for six quantifiers from Table 4.1. In the figure, vertical bars represent

pre-registered point probability beliefs about corresponding quantifiers from Table 4.1.

We note that while a few estimates (e.g., 'rarely' and 'often') roughly align with pre-registered beliefs, others are somewhat off (e.g., 'likely' shows a much higher value, partly since it usually occurred prefixed by 'most' in the data) and yet others (e.g., 'sometimes') show a large spread of values (indicating their diffused semantics) to be meaningfully modeled as point values. LNQ's performance, inspite of this, shows strong stability in the approach.

We don't use these empirically observed probabilities in experiments, (instead of pre-registered values), so as not to tune the hyperparameters to a specific dataset. Such estimates would not be available for a new task without labeled data. More importantly, using labeled data for estimating these probabilities, and then using the learned model for predicting labels would constitute overfitting, biasing evaluation.

4.5 Discussion

In this chapter, we made a simple proposal, arguing for leveraging declarative language for learning classifiers without labeled examples. In general, language can convey an expansive range of declarative knowledge. However, our exploration suggests that a large fraction of concept explanations in everyday natural language can be converted to quantitative constraints about statistical expectations about the true model. In particular, a lot of such explanations can effectively be thought of as fuzzy conditional rules about features and concept labels, where *fuzziness* can be quantified as a conditional probability. Leveraging natural language quantifiers, which are frequent denoters of generality in everyday language, can be an effective way to model this knowledge.

Our approach for learning from natural language quantification is surprisingly effective. However, it does not address linguistic issues such as modifiers (e.g., *very* likely), nested quantification, etc. On the other hand, we found no instances of nested quantification in

the data, suggesting that people might be primed to use simpler language when teaching. While we approximate quantifier semantics as absolute probability values, they may vary significantly based on the context, as shown by cognitive studies such as Newstead and Collis (1987). However, the relative stability of our approach to changes in probability values suggests that the learning procedure is encouraging in this regard. A notable aspect of the training procedure (also mentioned previously) framing the PR optimization problem with slack variables (thus allowing violations) allows for learning in scenarios with presence of contradictory advice.

Future work can model how these parameters can be adapted in a task specific way (e.g., cases such as cancer prediction where base rates are small), and provide better models of quantifier semantics. e.g., as distributions, rather than point values. Another important exploration might focus on modeling the semantics of *default quantification* of conditional sentences, i.e. model the degree of generality of explanations without an explicit quantifier (e.g., '*emails from my boss are important*'). Our proposed approach currently ignores such explanations. The meaning of such statements can depend on pragmatic or task-specific cues, which future directions can explore.

Perhaps even more importantly, our exploration of the problem also does not consider the issue of *trustworthiness* of a teacher. In settings where a learner learns from multiple teachers (e.g., consider an agent trying to learn to identify people who are at risk of cancer by reading advice from the web), some teachers may be more reliable than others. Ideally, an agent would also learn whom to trust over time, rather than weighing all advice equally.

Nonetheless, our approach in this chapter is a step towards the idea of using language as supervision to guide statistical learning. Inductive bias through language can enable learning in scenarios with limited or no labeled data, where traditional learning methods struggle. The main contributions of this work are:

1. We present an approach for training classifiers from natural language advice, which

can alleviate the need for labeled data for classification (and potentially require no labeled examples).

2. We develop datasets for zero-shot classification from natural descriptions, exhibiting tasks with various levels of difficulty.
3. We empirically show that coarse probability estimates to model linguistic quantifiers can effectively supervise model training across three domains of classification tasks.

Chapter 5

Jointly learning Concept Classification and Semantic Parsing with Weak Supervision

In the previous chapters, we explored two mechanisms through which interpreted language can help concept learning: by specifying useful features for a learning task, and by providing quantitative constraints on the concept model to be learned. However, both these approaches presume the presence of competent semantic parsing models that can interpret free-form natural language. In this chapter and the next, we provide new methods that focus on the complementary problem of *learning to interpret* i.e. learning the mapping from natural language statements to their correct interpretations. In particular, we provide new algorithms for learning semantic parsers that incorporate different types of situational context in the process of language interpretation.

In this chapter, we extend the setting in Chapter 3 and provide an approach for jointly learning both a semantic parser and a concept model, where the supervision consists only of labeled examples of the concept. The model does not have access to the logical form annotations for a statement at any point. Rather, the interpretations of statements are

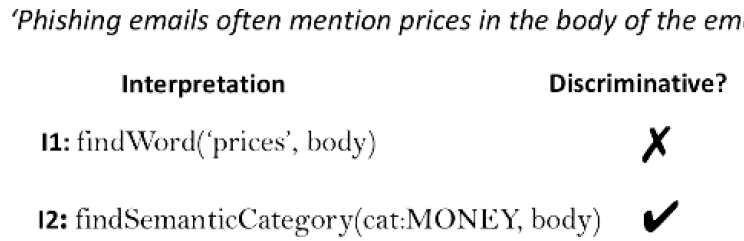


Figure 5.1: A semantic parser may associate multiple logical forms with a statement. Among these, correct interpretations are much more likely to help in discriminating instances of the concept. We can leverage this idea to jointly learn a semantic parser and a concept classifiers only from natural language explanations and labeled examples of the concept

treated as *latent variables*, and the learning of the parsing model is driven only through performance on a downstream task of concept learning. The only supervision comes in the form of a small number of labeled examples of a concept to be learned; the method is automatically biased towards learning interpretations of explanations that are helpful for the learning task (see Figure 5.1).

Experiments with our approach show that *sensory context* (in the form of feature value observations) can guide semantic parsing models towards correct language interpretations. This is a significantly weaker form of supervision than has been previously used for training semantic parsers. Semantic parsers have traditionally been trained on data consisting of statements paired with logical forms or their denotations. In our approach, even denotations are not directly observed, but inferred as latent variables.

Work described in this chapter has previously appeared in Srivastava et al. (2017b).

5.1 Approach

We address the task of learning concepts from natural language statements and a small number of labeled examples of the concept. Figure 5.2 summarizes the outline of our approach. Similar to Chapter 3, our approach maps statements to logical interpretations, which can be evaluated in context of new instances. In doing this, each statement s

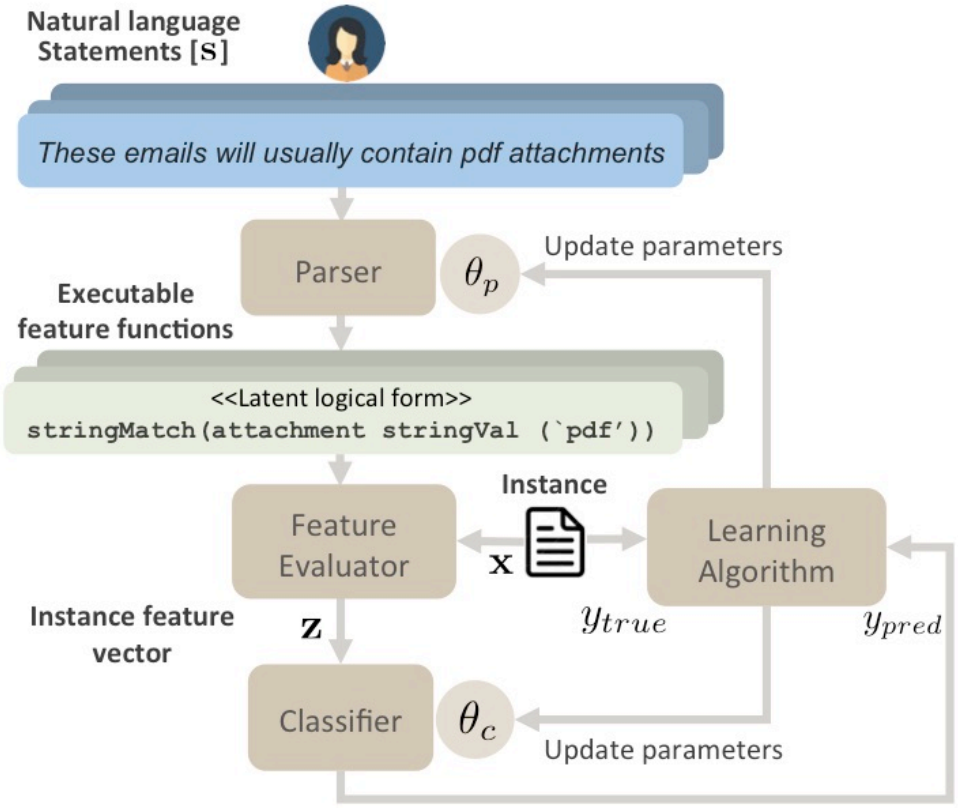


Figure 5.2: Schematic representation of joint approach for learning both a semantic parsing model (with parameters θ_p) and a concept classification model (with parameters θ_c). The only supervision for the approach is a set of labeled examples of the concept. As in Chapter 3, natural language explanations are mapped to logical forms that define feature functions. Over time, the model learns better interpretations of natural language statements, which lead to better feature representations for the concept learning problem

effectively acts as a binary feature function $\{z = f_s(x) \in \{0, 1\}\}$ that fires when the interpretation of a statement s is true for an instance x. However, here we show that is also possible to jointly learn a semantic parsing model with no additional supervision. The crux of our approach is that correct interpretations of natural language explanations are more likely to be useful in discriminating concepts, and this observation can be used to jointly learn both a semantic parser and a concept classifier.

Next, we describe our probabilistic latent variable formulation that learns a semantic parser and a concept classifier from labeled examples of the concept. The latent variables correspond to evaluations of natural language statements for different instances, and training proceeds via a generalized EM procedure that iteratively (1) estimates evaluations of explanations (marginalizing over all interpretations), and (2) updates the classification and semantic parsing models. The inputs to the method consist of a small number of labeled examples and non-examples of a concept, natural language statements explaining the concept, and a domain specific lexicon. The method does not require labeling sentences with logical forms.

5.1.1 Problem setting

We consider concept learning problems in which the goal is to approximate an unknown classification function $f : X \to Y$ where $Y = \{0, 1\}$. The input to our learning algorithm consists of a set of labeled training examples $\mathcal{T} := \{(x_1, y_1), \ldots, (x_m, y_m)\}$, along with a set of natural language statements $\mathcal{S} := \{s_1 \ldots s_n\}$ about the concept. Our aim is to leverage statements in $\mathcal{S}$ to learn a better classifier for the concept. Our training data does not contain any other form of supervision (such as logical forms).

We assume that each statement s_j defines some Boolean property over the instances X; that is, statement s_j should be interpreted as defining a predicate $l_j : X \to \{0, 1\}$. We augment the representation of each instance, x_i, with a feature vector $\mathbf{z}_i$, that encodes the

	s_1	s_2	s_j	s_m	Label
x_1	z_{11}	z_{12}	z_{1j}	z_{1m}	y_1
...					...
x_i	z_{i1}	z_{i2}	z_{ij}	z_{im}	y_i
...					...
x_n	z_{n1}	z_{n2}	z_{nj}	z_{nm}	y_n

Figure 5.3: Our data consist of instances x_i with binary labels y_i and statements $s_1 \dots s_n$ about a concept. z_{ij} denotes whether the statement s_j applies to instance x_i, and is not observed in the data. z_{ij} are not observed directly: they are computed by parsing s_j to a logical form l_j, and evaluating it in context of an instance x_i, i.e. $z_{ij} = [\![l_j]\!]_{x_i}$. We depart from Chapter 3 in not assuming supervision for training a parsing model. Rather, both the statement interpretations l_j and their evaluations z_{ij} are treated latent variables

information contained in $\mathcal{S}$. The individual elements of this feature vector, $z_{ij} \in \{0, 1\}$, denote whether the statement s_j applies to instance x_i (see Figure 5.3). In the general case, the evaluation values $\mathbf{z}_i$'s are not directly observed. These are obtained by parsing each statement s_j into a logical expression $l_j : X \to \{0, 1\}$ which can be evaluated for an instance x_i to obtain $z_{ij} = [\![l_j]\!]_{x_i}$.

In this paper, we jointly learn a classifier and a semantic parser while treating z's as latent variables. For training, we maximize the conditional log-likelihood of the observed data. Let us consider the log-likelihood for a single data instance (ignoring the subscript i) for now. Since the evaluations $\mathbf{z}$ of natural statements for any context are latent, we marginalize over these. Using Jensen's inequality, any distribution q over the latent variables provides a lower-bound on the data log-likelihood:

$$
\begin{aligned}
&\log p(y \mid x, \mathcal{S}) = \log \sum_{\mathbf{z}} p(y, \mathbf{z} \mid x, \mathcal{S}) \\
&\geq \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(y, \mathbf{z} \mid x, \mathcal{S})}{q(\mathbf{z})} \\
&= \sum_{\mathbf{z}} q(\mathbf{z}) \big(\underbrace{\log p_{\theta_c}(y \mid \mathbf{z}, x)}_{\text{classification}} + \underbrace{\log p_{\theta_p}(\mathbf{z} \mid x, \mathcal{S})}_{\text{parsing}} \big) \\
&\qquad\qquad + \mathcal{H}_q
\end{aligned}
\tag{5.1}
$$

Here, $\mathcal{H}_q$ is the entropy term for the distribution q.

5.1.2 Coupling parsing and classification

In Equation 5.1, we observe that the data likelihood decouples into the log probability of observing the concept labels $p_{\theta_c}(y_i \mid \mathbf{z}, x)$ conditioned on the statement evaluations and the log probability of the latent statement evaluations $p_{\theta_p}(\mathbf{z} \mid x, \mathcal{S})$. In particular, the first term can be naturally parameterized by a discriminative classifier such as a loglinear model (with associated parameters θ_c). We provide more details in Section 5.1.4.

On the other hand, the probability of the latent statement evaluation values $\mathbf{z}$ can be parameterized using a probabilistic semantic parsing model (with associated parameters θ_p). The second term decouples over evaluations of individual statements.

$$
\log p_{\theta_p}(\mathbf{z} \mid x, \mathcal{S}) = \sum_j \log p_{\theta_p}(z_j \mid x, s_j) \tag{5.2}
$$

In turn, since we never observe the correct interpretation l for any statement, but only model its evaluation z_j, we marginalize over all interpretations whose evaluations in a context x matches z_j (similar to Liang et al. (2011)).

$$\log p_{\theta_p}(z_j \mid x, s_j) = \log \sum_{l:[\![l]\!]_x = z_j} p_{\theta_p}(l \mid s_j) \tag{5.3}$$

Following recent work in semantic parsing (Liang and Potts, 2015; Krishnamurthy and Mitchell, 2012), we use a log-linear model over logical forms:

$$p_{\theta_p}(l \mid s) \propto \exp({\theta_p}^T \psi(s, l)) \tag{5.4}$$

where $\psi(s, l) \in \mathbb{R}^d$ is a feature vector over statements s and logical interpretations l.

5.1.3 Learning

In Equation 5.1, $q(\mathbf{z})$ denotes a distribution over evaluation values of statements; whereas θ_c and θ_p denote the model parameters for the classifier and semantic parser. The learning algorithm consists of an iterative generalized EM procedure, which can be interpreted as a block-coordinate ascent in the estimates of statement evaluations $q(\mathbf{z})$ and the model parameters θ_c and θ_p.

E-step: In the E-step, we update our estimates of evaluation variables ($\mathbf{z}$). We make a mean-field approximation by assuming that the joint distribution over evaluations decouples as $q(\mathbf{z}) = \prod q_j(z_j)$. Then maximizing the lower bound in Equation 5.1 in terms of q_j leads to the following update:

$$q_j(z_j) \propto \exp\Big(\mathop{\mathbb{E}}_{j' \neq j} [\log p_{\theta_c}(\mathbf{z}|x)] + \log p_{\theta_p}(z_j|x, s_j) \Big) \tag{5.5}$$

The first term in the update prefers values of an evaluation variable that are more discriminative on average (when values of other statements are marginalized out). The second term favours values of the evaluation variable that conforms with the most likely interpretations of the corresponding statement (s_j) by the semantic parser. Thus, in the E-step,

we upweight evaluations of statements that are both discriminative, as well as supported by interpretations from the semantic parser.

M-step: In the M-step, we update the model parameters to maximize the lower bound in Equation 5.1. This corresponds to independently optimizing the log-likelihood for the classification model and the semantic parser, based on current estimates of $q_j(z_j)$'s of the statement evaluations. The entropy term H_q is constant from the perspective of model parameters, and is not relevant for the optimization. In particular, the semantic parser is updated to agree with evaluations of natural language statements that are discriminative. At the same time, the classification model is updated to fit evaluations that are supported by interpretations from the semantic parser.

We now describe the M-step updates for the log-linear semantic parser with parameters, θ_p. The updates for the classifier parameters, θ_c, depend on the form of the classification model, and are described in Section 5.1.4. For clarity, we focus on updates corresponding to a particular statement s_j from the training dataset. From Equations 5.1, 5.3 and 5.4, the objective for the semantic parser is given by:

$$\ell_j(\theta_p) = \sum_i \sum_{z \in \{0,1\}} q(z_{ij}) \log \frac{\sum\limits_{l:[\![l]\!]_{x_i}=z} \exp({\theta_p}^T \psi(s_j, l))}{\sum\limits_{l} \exp({\theta_p}^T \psi(s_j, l))} \tag{5.6}$$

Semantic parsers are usually optimized using gradient updates. Here, the gradient is:

$$\nabla \ell_j(\theta_p) = \sum_{i,z,l} q(z_{ij}) p_{\theta_p}(l \mid s) \frac{p_{\theta_p}(z_{ij} \neq z | x_i, s)}{p_{\theta_p}(z_{ij} = z | x_i, s)} \psi(s_j, l) \tag{5.7}$$

5.1.4 Classification models

The problem formulation and learning procedure described in Sections 5.1.2 and 5.1.3 is agnostic to the choice of the classification model (with parameters θ_c). For this work, we experimented with a logistic classifier (LR) and a Naive Bayes model (NB).

Logistic Regression (LR)

The form of the logistic function $\log p(y|\mathbf{z}) = -\log(1 + \exp(\text{-}\theta_c^T \mathbf{z}\ y))$ means that the likelihood does not decouple for individual components in $\mathbf{z}$. Hence, in the E-step, the expectation in Equation 5.5 cannot be computed analytically. Instead, we estimate this by drawing Bernoulli samples for individual z_j's using previous estimates of $q_j(z_j)$. In the M-step, we update classification parameters θ_c using stochastic gradient updates, while again sampling individual z_j's.

Naive Bayes (NB)

The likelihood for this model is $p(y, \mathbf{z}) = \prod_j \theta_{cy}^{z_j}(1 - \theta_{cy})^{1-z_j}$. In this case, the individual components of $\mathbf{z}$ decouple in the log likelihood, leading to simple updates in both the E and M steps. While this is not a conditional likelihood (as expected in Section 5.1.2), in our experiments we found that the NB objective to be empirically effective with our approach.

5.2 Data

For evaluation of our approach, we use the data previously described in Section 3. The data consists of a collection of 1037 emails belonging to one of seven email categories, and 235 natural language explanations describing these categories. The explanation statements are also annotated with logical forms. As opposed to our approach in Chapter 3, the joint approach presented here does not use these labeled logical forms to train a semantic parser.

	Contact	Employee	Event	Humor	Meeting	Policy	Reminder	Average
Joint LNL(LR)	0.608*	0.351	0.568*	0.570*	0.757*	0.898*	0.437	0.598*
Joint LNL(NB)	0.628*	0.370	0.453*	0.590*	0.732*	0.878*	0.414	0.581*
LNL	0.648*	0.381	0.627*	0.565*	0.770*	0.892*	0.470	0.622
LNL-Gold	0.661	0.397	0.677	0.572	0.777	0.917	0.487	0.641
Joint LNL(LR) + BoW	0.634	0.398	0.604	0.704	0.747	0.891	0.567	0.649
Joint LNL(NB) + BoW	0.644	0.409	0.520	0.709	0.723	0.878	0.543	0.632
LNL+BoW	0.640	0.421	0.622	0.755	0.731	0.909	0.554	0.661
LNL-Gold+BoW	0.667	0.449	0.659	0.798	0.771	0.927	0.595	0.695

Table 5.1: Concept learning performance (F1 scores) using $n = 10$ labeled examples for *Joint LNL (LR)* and *Joint LNL(NB)* (reproduces and extends results from Table 3.4). * for the rows corresponding to LNL denotes statistical significance over the best performing non-LNL model

Rather, the supervision is only through concept labels for email instances provided to the method. However, these annotations are used to evaluate the semantic parsing performance of the approach.

5.3 Evaluation

In this section, we evaluate the performance of our approach from the perspectives of concept learning as well as semantic parsing. We first compare our methods against traditional text classification methods on the task of email classification (the baseline models are the same as from Section 3.4).

5.3.1 Concept Learning

Table 5.1 reproduces results from Table 3.4 from Chapter 3, but also includes classification performance of our joint approaches for concept learning and semantic parsing. These are denoted by *Joint LNL(LR)* and *Joint LNL(NB)* in the table, which correspond to models with a Logistic Regression and a Naive Bayes parametrization for the classifier component. As before, the models are trained for $n = 10$ labeled examples, and the reported numbers are averages of F1 scores over 10 uniform random draws of the labeled examples from the

data set of emails.

We observe that both *Joint LNL(LR)* and *Joint LNL(NB)* (Logistic Regression) outperform all baselines approaches, and are consistently only slightly lower in performance than the fully supervised LNL approach from Chapter 3. Interestingly, we also note that *Joint LNL(LR)* and *Joint LNL(NB)* show similar performance for most concepts. As in Table 3.4, *LNL-Gold* denotes the classification performance with using these annotated gold parses. This corresponds to the hypothetical case where the classifier knows the correct semantic interpretation of each natural language sentence from an oracle. The last four rows in the table show performance when the *LNL* methods also utilize BoW representations of the data. These show consistent gains in overall performance, while conforming to the same pattern.

These results indicate that our weakly supervised method is quite effective in interpreting natural language statements for concept learning, without explicit supervision for logical forms.

5.3.2 Semantic Parsing

We next evaluate the parsing performance of our approach, which learns a semantic parser from only concept labels of examples. Table 5.2 evaluates parsing performance against the gold annotation logical forms for statements. For this task, we check for exact match of the predicted logical forms with manually annotated logical forms for the concept explanations. In the table, full supervision refers to traditional training of a semantic parser using complete annotations of statements with their logical forms, which is the approach we previously follow in Chapter 3. The results report average accuracy over 10-fold cross-validation, and demonstrate that while not comparable to supervised parsing, our weakly supervised approach is relatively effective in learning semantic parsers.

Further, exact match to gold annotated logical forms is a restrictive measure. Qualita-

	Accuracy
Fully Supervised	0.63
Joint LNL (LR)	0.30
Joint LNL (NB)	0.28
No training	0.01

Table 5.2: Semantic parsing performance (exact match) for proposed weakly supervised methods (*Joint LNL(LR)* and *Joint LNL(NB)*) vs full supervision (completely labeled logical forms)

tive analysis revealed that even when the predicted and gold annotation logical forms don't match, predicted logical forms are often strongly correlated in terms of evaluation to gold annotations. e.g., `getPhraseMention( email, stringVal(`postdoc'))` vs `getPhraseMention(body, stringVal(`postdoc'))`. In about 5% of cases, predicted and gold interpretations are different on the surface, but are semantically equivalent (e.g., `stringEquals( sender, recipient)` vs `stringEquals( recipient, sender)`).

5.4 Discussion

We consider the problem of jointly learning a semantic parser and a concept classifier from natural language explanations, and a small number of labeled examples of the concept. On email classification tasks, this approach yields improvements in classification performance comparable with an approach that trains a fully supervised model for mapping explanations to feature functions.

More significantly, this scenario demonstrates an application where sensory context from the environment (in the form of feature value observations) in conjunction with concept labels can be used to weakly supervise the training of a semantic parser, in absence of usual semantic annotations (sentences paired with labeled logical forms or denotations of logical forms). Our approach leverages pragmatics to prefer interpretations of natural language explanations that are more discriminative in context of concept learning. Our empirical results show that such context may be a viable and inexpensive source of su-

pervision for training semantic parsers. Further, a parsing model trained using such an approach (using natural language explanations and labeled examples of a concept) can subsequently be used to bootstrap training of classifiers for other concepts where there are no labeled examples (such as using the approach presented in Chapter 4).

Our contributions are:

- We provide a joint method for concept learning and language understanding that can be trained from a small number of labeled concept instances alone.
- Our empirical results show that such context may be a viable and inexpensive source of supervision for training semantic parsers. We leveraged pragmatics to prefer interpretations of natural language explanations that are more discriminative in context of concept learning. In doing this, we extend supervised semantic parsing by learning from a weaker form of supervision than learning from denotations. This is a significantly weaker setting than either learning from logical forms, or learning from denotations.
- We demonstrate that for small labeled data, such an approach can achieve substantial gains in classification accuracy by using natural language explanations, even without a fully supervised parsing model.

Chapter 6

Semantic Parsing with Conversational Context

Teaching machines through natural conversation involves significant challenges for semantic parsing. Most extant methods for semantic parsing focus exclusively on parsing single natural language sentences in isolation. While this maybe useful for environments consisting of one-shot interactions of users with a system (e.g., running QA queries on a database), conversational settings often require extended interactions between a user and an automated assistant, where a model of previous conversational history maybe fundamental for understanding the meaning of a new statement from a user.

We illustrate this in Figure 6.1, which shows a snippet of a real conversation between a human user and an automated email assistant. In this conversation, the user attempts to instruct the assistant to perform a sequence of tasks related to sending an email. From the figure, we note that pragmatics and conversational context are fundamental for understanding several individual utterances from the user. In particular, observe that utterance U2 (*'Set to mom's email'*) cannot be correctly parsed based on its content alone, but requires an understanding of the discourse. Based on the previous statement (composing a new email), setting the recipient field is the user's likely intent. Similarly, utterances U3

```
[U1] User: Compose an email
[A1] Agent: Composing new email. Its fields are: recipient, subject and body
[U2] User: Use mom's email
[A2] Agent: I set the recipient field to `momthebest@email.com'
[U3] User: Set the subject to hello, please
[A3] Agent: I set the subject field to `hello, please'
[U4] User: Set the subject to hello, please
[A4] Agent: I set the subject field to `hello'
[U5] User: Set the body to I'll be late
[A5] Agent: I set the body field to `I'll be late'
[U6] User: Send the email
[A6] Agent: I sent the email ...
```

Figure 6.1: Example of a real-world interaction between a human (User) and an automated email assistant (Agent)

and U4 show an example of a repetition, where the agent first misinterprets (U3), and then correctly parses a statement (U4). While parsing U4, the agent needs to implicitly understand that it should interpret the current utterance to a different logical form than before (even though the textual content is identical). This would not be possible in the traditional one-shot parsing setting, which cannot incorporate such implicit feedback. Instead, correctly interpreting the sentence requires modeling of the discourse structure of the conversation.

In this chapter, we provide an approach for semantic parsing of natural language that incorporates such *conversational context*. In particular, we show that the state of a conversation immediately preceding a statement can provide contextual features that can improve semantic parsing performance. We incorporate such structural features for modeling the flow of discourse and context in a conversation, and provide a training algorithm for learning such a model from conversation snippets consisting of sequences of statements annotated with logical forms.

Work described in this chapter has previously appeared in Srivastava et al. (2017a).

6.1 Related Work

Supervised semantic parsing has been studied in a wide range of settings (Zettlemoyer and Collins, 2005; Wong and Mooney, 2007; Kwiatkowski et al., 2010). Recent approaches have focused on various strategies for using weaker forms of supervision (Clarke et al., 2010; Krishnamurthy and Mitchell, 2012; Berant et al., 2013) and rapid prototyping of semantic parsers for new domains (Wang et al., 2015; Pasupat and Liang, 2015). Other works have explored semantic parsing in a grounded contexts, and using perceptual context to assist semantic parsing (Matuszek et al., 2012; Krishnamurthy and Kollar, 2013). However, none of these approaches incorporate conversational context to jointly interpret a conversational sequence. For instance, poor interpretations made while parsing at a point in a conversation cannot be re-evaluated in light of more incoming information. A notable work that incorporates conversational data in relation to semantic parsing is (Artzi and Zettlemoyer, 2011). However, rather that incorporating contextual cues, their goal is very different: using rephrasings in conversation logs as weak supervision for inducing a semantic parser. Closer to our work is previous work by Zettlemoyer and Collins (2009), who also learn context-sensitive interpretations of sentences using a two-step model. However, their formulation is specific to CCG grammars and focuses on modeling discourse referents.

The role of context in assigning meaning to language has been emphasized from abstract perspectives in computational semantics (Bates, 1976; Van Dijk, 1980), as well as in systems for task-specific applications (Larsson and Traum, 2000). Examples of the former include analyzing language from perspectives of speech acts (Searle, 1969) and semantic scripts (Schank and Abelson, 1977; Chambers and Jurafsky, 2008; Pichotta and Mooney, 2015). These works induce typical trajectories of event sequences from unlabeled text to infer what might happen next.

On the other hand, a notable application area that has explored conversational context within highly specific settings is state tracking in dialog systems. Here, the focus is on

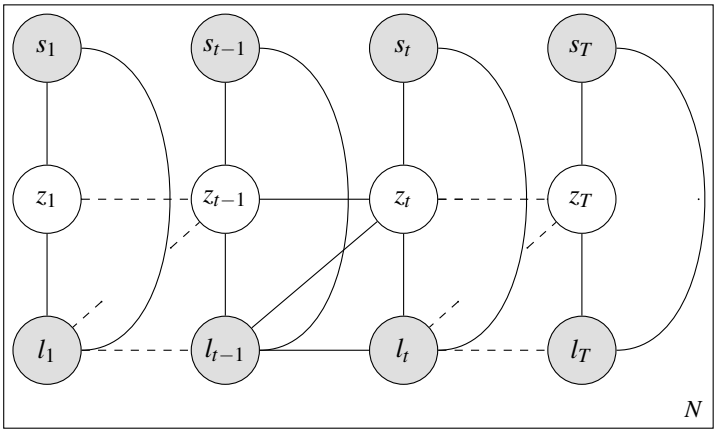


Figure 6.2: Model diagram for semantic parsing of conversational sequences. Traditional semantic parsing features depend on utterances s_t and associated logical forms l_t only. Our model additionally allows structured features that can depend on previous logical forms l_{t-1}, latent variables z_t representing the discourse state of the conversation at any step, and the previous utterances $s_1 \dots s_t$

inferring the state of a conversation given all previous dialog history (Higashinaka et al., 2003; Williams et al., 2013) in context of specific task trajectories, rather than interpreting the semantic meanings of individual utterances.

We allow our semantic parser to output logical forms that may not be entailed from an utterance using a strict grammar formalism, expanding on similar ideas in Wang et al. (2015); Goldwasser and Roth (2014). Finally, some of the heuristics proposed in this work are motivated by previous work on paraphrasing for semantic parsing (Berant and Liang, 2014).

6.2 Approach

We present an approach for semantic parsing of conversations by posing conversations as sequences of utterances to model 'flow of discourse'. We consider the problem as a structured prediction task, where we jointly learn preferences for collective assignments of logical forms for sentences in a sequence.

Let $\mathbf{s}$ denote a conversation sequence of T utterances by a user, with individual ut-

terances denoted as $\{s_1 \dots s_T\}$. Let $\mathbf{l} := \{l_1 \dots l_T\}$ be the intended logical forms for corresponding utterances. We assume a supervised learning setting where we have labeled training sequences $\mathcal{T} := \{(\mathbf{s}^{(1)}, \mathbf{l}^{(1)}) \dots (\mathbf{s}^{(N)}, \mathbf{l}^{(N)})\}$ consisting of utterances and their associated logical forms. In comparison, the traditional supervised setting for learning semantic parsers consists of pairs of training utterances and associated logical forms (s_i, l_i), but doesn't have a sequential structure. Our model utilizes this sequential structure to incorporate information about discourse and pragmatics.

In addition, we also associate a latent categorical variable denoted as z_t with each user utterance s_t to reinforce the modeling of the flow of discourse (see Figure 6.2). The latent states can take one of K possible discrete values, and abstractly represent distinct discourse states. The value of K is a predefined parameter for the model[1]. In Section 6.5, we show that these latent states learn distinct interpretable discourse states that are prevalent in conversations, and can support dynamic modeling of context with the progress of a conversation.

For a given utterance sequence $\mathbf{s} = \{s_1 \dots s_T\}$, our model predicts logical assignments, $\widehat{\mathbf{l}} = \{\hat{l}_1 \dots \hat{l}_T\}$, and latent discourse states, $\widehat{\mathbf{z}} = \{\hat{z}_1 \dots \hat{z}_T\}$ by solving the following inference problem, i.e. finding the highest scoring assignment of logical forms, $\mathbf{l}$, and discourse states, $\mathbf{z}$, under a given model:

$$(\widehat{\mathbf{l}}, \widehat{\mathbf{z}}) = \underset{\mathbf{l} \in \mathcal{L}(\mathbf{s}), \mathbf{z}}{\operatorname{argmax}} \quad S_{\theta_p}(\mathbf{s}, \mathbf{l}, \mathbf{z}) \tag{6.1}$$

Here, $\mathcal{L}(\mathbf{s})$ is the search space associated with sequence $\mathbf{s}$, consisting of possible joint assignments of logical forms to the various utterances in the sequence. For any utterance s_t, the grammar of the meaning representation formalism can specify the set $\mathcal{G}(s_t)$ of its candidate logical forms. The associated search space for the sequence $\mathbf{s}$ is then simply

[1]Setting $K = 1$ effectively reduces the model to not using latent variables at all. This model still incorporates structural context and discourse by learning preferences for joint assignments of logical forms to utterances. However, the latent variables z_t afford additional flexibility to the model. i.e. for the same utterance, the model can parse differently based on the current state

given by the cross-product $\mathcal{L}(\mathbf{s}) = \otimes_t \mathcal{G}(s_t)$, and the hat symbol ($\widehat{\ }$) represents predicted variables.

$S_{\theta_p}(\mathbf{s}, \mathbf{l}, \mathbf{z})$ represents a linear score denoting the goodness of an assignment of logical forms, $\mathbf{l}$, and latent discourse states, $\mathbf{z}$, to utterances in conversation sequence $\mathbf{s}$. This score is defined as:

$$S_{\theta_p}(\mathbf{s}, \mathbf{l}, \mathbf{z}) = {\theta_p}^T \psi(\mathbf{s}, \mathbf{l}, \mathbf{z})$$

where, ψ is a feature function that produces a real-valued feature vector for the tuple $(\mathbf{s}, \mathbf{l}, \mathbf{z})$. As we shall see in Section 6.3, these features consist of two categories $\psi = [\psi_{text}\ \psi_{context}]$: (a) ψ_{text}: features for individual parses that model how well individual logical forms, l_t, match corresponding natural language utterances, s_t, in isolation. This subset subsumes all features from traditional semantic parsing models (b) $\psi_{context}$: features that model conversational context and discourse across the chain structure of the conversation. The model parameters, θ_p, consist of a real-valued weight for each kind of feature, and are learned during training.

6.2.1 Model training

The model parameters θ_p can be trained via the latent variable Structured Perceptron algorithm (Collins, 2002; Zettlemoyer and Collins, 2007), which performs subgradient updates minimizing the following structured hinge loss:

$$L(\theta_p, s, l, z) := \left| \max_{\hat{\mathbf{l}} \in \mathcal{L}(\mathbf{s}), \hat{\mathbf{z}}} S_{\theta_p}(\mathbf{s}, \hat{\mathbf{l}}, \hat{\mathbf{z}}) - \max_{\mathbf{z}^\star} S_{\theta_p}(\mathbf{s}, \mathbf{l}, \mathbf{z}^\star) \right|_+ \tag{6.2}$$

The objective consists of a difference of two terms: the first is the score of predicted assignment $(\widehat{\mathbf{l}}, \widehat{\mathbf{z}})$ for sequence $\mathbf{s}$ under the current model, while the second is the score of the highest-scoring latent discourse states for $\mathbf{s}$ with the ground truth logical form $\mathbf{l}$. These

correspond to solving the following inference problems: (1) finding the best combination of logical forms and discourse states for a sequence (Equation 6.1), and (2) finding the best combination of discourse states for a sequence and given logical forms. We describe the procedure to solve Equation 6.1 below. The second inference problem is a simpler case of the same equation, since one of the two variables to be inferred ($\mathbf{l}$) is already known. Finally, in our experiments, we also use an $l2$-regularizer (with ridge parameter 0.01) for weight-vector θ_p.

We note that our formulation does not pre-suppose a specific semantic parsing framework, grammar or feature-definition. In this work, we use a CCG-based semantic parsing approach. However, our framework can seamlessly extend to other formalisms such as DCS (Liang et al., 2013) that are trained with gradient updates.

6.2.2 Inference

Both training and prediction for the model depend on efficiently solving the inference problem in Equation 6.1. In general, the problem can be tractably solved if components of feature function ψ decompose into smaller factors. In our case, ψ_{text} features decompose according to the structure of individual parse trees. Similarly, $\psi_{context}$ features factorize according to chain structure of the discourse due to Markov properties (details in Section 6.3). Our inference procedure consists of a hierarchical two step process: (1) we find a candidate set of possible logical forms for each utterance, s_t, in a sequence, and (2) we find the best joint assignment among these by incorporating information from contextual and structural cues. We now briefly describe the two steps.

In the first step, we obtain a set of candidate logical forms from the semantic parsing grammar, $\mathcal{G}(s_t)$, for individual utterances, s_t, while viewing them in isolation. To prune this set to a manageable size, we score a potential logical form using only the text-based

features (ψ_{text}). This is identical to traditional semantic parsing, and the highest scoring logical forms for an utterance can be found using the k-best CYK algorithm. In practice, considerations such as large grammars make exact inference prohibitive for this setting. Following previous works in semantic parsing (Kwiatkowski et al., 2013; Berant et al., 2013), we employ beam search to find an approximate set of best candidate logical forms for a sentence.

In the next step, we combine the above-obtained sets of candidate logical forms for individual utterances, s_t, to infer the best joint semantic parse $\mathbf{l}$ (and discourse states $\mathbf{z}$) for the complete sequence, $\mathbf{s}$. This involves obtaining a sequence of l_t's that incorporates scores from the contextual and discourse features ($\psi_{context}$). Since these features decompose according to the chain structure of the sequence, the highest scoring assignments of a sequence of discourse states $\mathbf{z}$ and logical forms $\mathbf{l}$ can be efficiently computed using the Viterbi algorithm (where *hidden* states of the Viterbi chart correspond to pairs of logical forms in the candidate set and discrete discourse states).

6.2.3 Expansion strategies

The approach described above uses the traditional semantic parsing setting (using the score from ψ_{text} only) to define the set of candidate logical forms for an utterance, $\mathcal{G}(s_t)$. However, one may define strategies to expand the candidate set $\mathcal{G}(s_t)$ to also include logical forms that are not included in the beam from the text-only features. We consider the following simple heuristics to expand the candidate set, $\mathcal{G}(s_t)$, for each utterance s_t:

1. **Highest PMI (PMI)**: Add to $\mathcal{G}(s_t)$ the logical forms that have the n-highest PMI with the best scoring logical form from the text-only model for the previous utterance (s_{t-1}) in the training set.
2. **Highest conditional probability (Prob)**: Add to $\mathcal{G}(s_t)$ the n most frequent logical forms that followed the best scoring logical form from the text-only model for

the previous utterance (s_{t-1}) in the training set.

3. **Paraphrase (PP)**: Add to $\mathcal{G}(s_t)$ the candidate sets for k utterances in the training set that are semantically most similar to s_t. For computing similarity, we use Vector Tree Kernels (Srivastava et al., 2013; Srivastava and Hovy, 2014) that provide a semantic similarity score between two sentences using syntactic information and distributional embeddings.

4. **Most frequent (MF)**: Add the n-most frequent logical forms observed in the training data to the candidate set $\mathcal{G}(s_t)$ for each utterance.

6.2.4 Prediction

Our model is trained to simultaneously consider all utterances of a sequence. At prediction time however, in most scenarios, the agent would need to continually infer the parse of the latest sentence during the conversation, in a real-time setting. In particular, we need to ensure not to use future utterances while predicting the logical form for the current utterance. Hence, at prediction time, we simply use the model to predict logical forms for the sequence of conversation ending at the current utterance.

6.3 Features for Incorporating Conversational Context in Semantic Parsing

In this section, we outline the text-based and context-based features used by our model. The text-based features are based on lexical and grammar rules, typically employed in traditional semantic parsers, whereas the context-based features are based on simple counts of specific configurations of logical predicates and discourse state assignments for a sequence. Thus, both kinds of features can be computed efficiently.

6.3.1 Text-based features

These features (denoted by ψ_{text}) depend on a single utterance s_t and a candidate logical form l_t. As such, they lie in the ambit of traditional semantic parsing features, and are standard features for CCG semantic parsers (Zettlemoyer and Collins, 2007; Azaria et al., 2016; Artzi and Zettlemoyer, 2013). We use the following text-based features:

- Lexicon features: Indicator features for each lexicon entry that fires for the given utterance, and indicator features for each syntactic category (POS) in the utterance combined with the logical form.
- Rule application features: Indicator features for both unary and binary rules in the parse of the given utterance to the associated logical form.
- String-based features: Number of words in the utterance, indicator features denoting whether string spans occur at the beginning or end of the utterance.

6.3.2 Structural context-based features

These features (denoted by $\psi_{context}$) model the flow of discourse and context-specific regularities by learning preferences for correlations between logical predicates, discourse state assignments and features based on the text of the conversational history.

- Transition features: Indicator features denoting combinations of logical predicates in successive utterances (e.g., $\{L_i$:`setBodyToString`, L_{i-1}:`createEmail`$\}$)[2], combinations of discourse variable assignments in successive utterances (e.g., $\{Z_i$=`State1`,Z_{i-1}=`State2`$\}$), combinations of logical predicates and discourse variable in successive steps.
- Emission features: Indicator features denoting presence of a logical predicate in the current utterance combined with the current discourse state (e.g. $\{Z_i$=`State0`,L_i:`greeting`$\}$).
- Lexical trigger features: Indicator features denoting presence of trigger words (from

[2] $\{L_i$:`a`,L_j:`b`$\}$ denotes that the logical form L_i includes the logical predicate `a`, and L_j contains `b`.

CCG lexicon) in the current utterance, paired with logical predicates in the current logical form and the current discourse state.

- Repetition features: Indicator features denoting whether the current utterance is a repeat, Indicator features denoting if the current utterance is a repeat and the current logical form is the same as the previous step, etc.[3]
- Domain-specific features: Indicator feature that looks at long term history to denote whether the user is currently teaching the system a new procedure (e.g., `inProcedure=true`). See Section 6.4 for explanation.
- Positional features: Feature denoting the position of the current utterance in the conversation.
- Provenance features: Indicator feature denoting whether the current logical form was derived from the CCG grammar, or added through an expansion strategy.

6.4 Data

Most existing datasets for semantic parsing focus on understanding single utterances at a time, rather than conversations. Thus, we created a dataset of real-life user conversations in an email assistant environment. For this, we annotated transcripts of real conversations between human subjects and an email assistant agent in a text-dialog environment provided in previous work by Azaria et al. (2016).

Each interaction session consists of the user trying to accomplish a set of email-based tasks by interacting with the agent (similar to those seen in Figure 6.1). The system also allows users to teach new procedures (e.g., forwarding an email), concepts (e.g., concept of a contact with fields name, phone number, etc.), and instances (e.g., instantiating a contact) on-the-fly. Because of this feature, and since the users are unaware of the capabilities of

[3]While Figure 6.2 indicates possible edge features between latent variables (z_t or l_t) and s_t , the model also allows features that could depend on the entire history of utterances observed till t ($s_1 \dots s_t$).

the agent, linguistic usage in the experiments is complex and diverse, compared to many existing datasets. The data consists of sequences of user utterances and system responses. In order to make the data usable for research, we pruned conversation sequences and annotated user utterances with their associated logical forms. e.g., the utterance: *'What is Mom's email?'* is annotated with the logical form `(evalField (getFieldByInstanceName mom email))`, following the logical language in the original paper (Azaria et al., 2016). Utterances that could not be reasonably expected to be interpreted by the email agent were marked as `unknownCommand` (7% of utterances). However, if the user later taught the system what she meant, future instances of the utterance were marked with the intended logical form (e.g., users often taught the command *'Next'* to read and move to the next email in the inbox). Sequences devolving into non-meaningful interactions were removed, e.g., if the annotator deemed that the user did not intend to complete a task. Superfluous segments of the original conversation (e.g., utterances re-phrasing a previous utterances that the system didn't process) were also manually pruned.

Annotating every command by manually specifying its logical form would require experience with the underlying logical language of the system. Instead, we developed a software that allowed faster annotation using an alternate procedure. The software allows annotators to load a conversation sequence, and execute each utterance against a live version of the email agent. If the response indicates that agent has correctly interpreted the command, the annotator may save the associated logical form for the utterance. However, if the response indicates that the system did not interpret the command correctly (judged by the annotator's belief of the user's intent), the annotator may provide a command which (i) reflects the intention of the utterance, and that (ii) the email agent can interpret correctly. In effect, the strategy uses annotators to paraphrase the original command into simpler commands (still in natural language) that the agent would parse to the correct logical form, without exposing them to the underlying meaning representation formalism. Thus, while this procedure still requires the annotator to be familiar with the capabilities of the email

Table 6.1: Corpus statistics for Email Assistant dataset

Number of user utterances	4759
User sessions	113
Avg length of session (utterances)	42
Word types (all utterances)	704

agent, the annotator is not required to know the specifics of the meaning representation language and logical predicates used by it.

Figure 6.1 summarizes the statistics for the curated dataset. For evaluation and future comparisons, we split the data into a training fold (93 conversation sequences) and a test fold (20 conversation sequences). The data described here is available at `http://www.cs.cmu.edu/~shashans/resources/conversationparsing`.

6.5 Evaluation

In this section, we discuss quantitative and qualitative evaluation of our method. We first make a comparative evaluation of our method in recovering gold-standard annotation parses on the held-out test set. Next, we make an ablation study to assess the contributions of different families of structural features described in Section 6.3. We then briefly analyze the characteristics of the latent states learned by the model, and qualitatively discuss the performance of the model.

6.5.1 Parsing performance

For training our models, we tune parameters, i.e. number of training epochs (5), and the number of clusters ($K = 3$) through 10-fold cross-validation on the training data. For the CCG grammar, we use the PAL lexicon induction algorithm (Krishnamurthy, 2016) to expand the base lexicon provided by Azaria et al. (2016). Our baselines include the

	Accuracy
Previous methods	
Unstructured CCG	51.9
Seq2Seq	52.3
LEX	46.4
Proposed models	
SPCon	54.2
SPCon + PMI	56.2
SPCon + Prob	56.9
SPCon + MF	**62.3**
SPCon + PP	59.8

Table 6.2: Test accuracies on Email Assistant dataset

following semantic parsing models:

- *Unstructured CCG:* CCG parser from Azaria et al. (2016), which uses the same lexicon and text-based features from Section 6.3, but does not incorporate structural features
- *Seq2Seq:* Deep neural network based on sequence-to-sequence RNN model from Bahdanau et al. (2015) to directly maps utterances to logical forms.
- *LEX:* Alignment-based model that chooses best parse using lexical trigger scores only, with no syntactic rules (does not use provided lexicon).

Table 6.2 compares the performance of variations of our method for semantic parsing with conversational context (SPCon) with baselines on the held-out test set of conversational sequences. Parses are evaluated for exact match in logical forms, and reported results are averages over 5 runs. We observe that the *Seq2Seq* and *Unstructured CCG* models perform comparably, whereas *LEX* doesn't perform as well.

We find that our structured models (*SPCon* and its variations) consistently outperform the baseline models. Further, the expansion strategies suggested in Section 6.2 lead to consistent gains in performance. The improvement of *SPCon* over *Unstructured CCG*, and further improvements of expanded models over *SPCon* are statistically significant ($\alpha = 0.1$, McNemar's test). In particular, the expansion strategies that afford highest coverage (MF

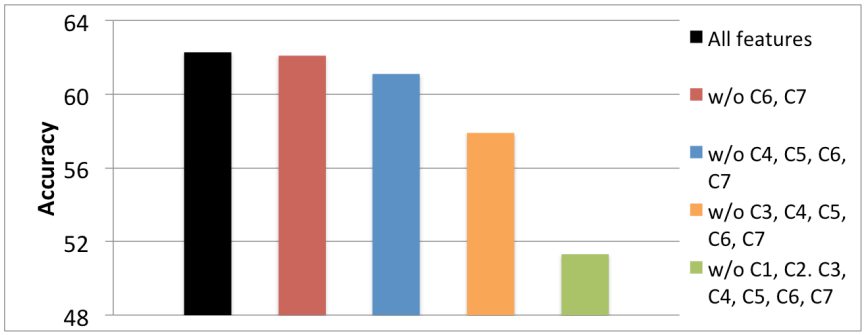


Figure 6.3: Comparison of parsing accuracy by successive removal of structured feature families described in Section 6.3. Removing structural transition and emission features (C1 and C2) leads to the most significant drop in performance

and PP) prove to be most effective. This suggests that the structural features are helpful for disambiguating between a large number of candidate logical forms, even when text-only features don't yield the correct logical form as a candidate. This also indicates that the performance of the unstructured CCG parser is restricted by issues of recall (the correct parse is not among the candidates from the beam search for the majority of error cases) due to the open-ended linguistic usage in the dataset. The expansion strategies partially alleviate this issue.

6.5.2 Feature ablation

Next, we perform an ablation study to analyze the contributions of various kinds of features to the model performance. Figure 6.3 shows the effects of successively removing different categories of structural features from the best performing model described above.

We note that removing positional and provenance features (C6 and C7) has minimal effect on model performance. Removing features identifying repetition and domain-specific features (C4 and C5) leads to a 1% drop. Further removing lexical trigger features (C3) that associate certain words with specific logical predicates and discourse states leads to a more significant drop. However, the biggest effect is seen by removing transition and

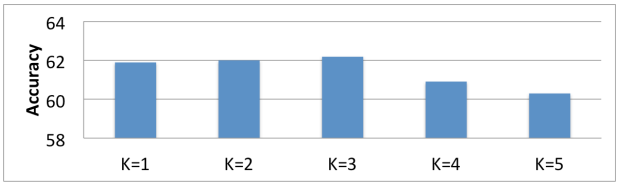


Figure 6.4: Semantic Parsing accuracy for different values of the number of latent discourse states, K. Setting $K = 1$ effectively reduces the model to not using latent variables at all. This model still incorporates structural context and discourse by learning preferences for joint assignments of logical forms to utterances. Larger values of K can enable additional modeling flexibility. i.e. for the same utterance, the model can parse differently based on the current state

emission features (C1 and C2). This is expected since these are fundamental for modeling associations across steps in the sequential structure of conversations. Ablating these features in the final step reduces the model to text-based features only, and the model performance is then understandably close to the performance of the unstructured CCG model (the marginal difference is due to batch updates for sentences in a conversation sequence vs online updates in the unstructured case). This also validates our thesis that incorporating contextual information leads to better models for understanding conversations.

6.5.3 Interpretations of Latent states

We investigate the effect of latent states on model performance. We varied the number of latent states (K) and found model performance deteriorated for more than $K = 3$ states (see Figure 6.4). We qualitatively explored contents of individual latent states to see if learned latent states reflect distinct discourse states in conversations. Table 6.3 characterizes some of the highest weighted features associated with each state for a run of the model. State 1 appears to be associated with confusion as it has highest weights for features that indicate presence of logical predicates `unknownCommand` and `cancel`. Similarly, State 2 is associated with teaching new procedures (the feature `inProcedure=true` has a high weight, and another high-weighted feature indicates presence of the logical predicate

State 1	has(`unknownCommand`), has(`cancel`)
State 2	`inProcedure=true`, has(`doSeq`)
State 3	has(`createInstanceByName`), has(`readInstance`)

Table 6.3: High-weight features for each latent state ($K = 3$). has(`a`) denotes a feature associated with the presence of logical predicate `a`

`doSeq` which is strongly associated with teaching procedures). On the other hand, State 3 has a more generic character, consisting of the most common logical predicates, and is the predicted state for the majority of utterances.

We also observe that latent states enable our approach to model interesting context-specific linguistic usage. For example, an analysis of state-specific weights learned by the model showed that it learns two distinct interpretations for the trigger word *cancel* in different contexts: within a learning procedure *cancel* is strongly associated with a logical predicate to quit the procedure, outside this context it is strongly associated with undoing the action taken in the previous step.

Errors: We found that in many examples, structured features partially address many of the issues highlighted in Figure 6.1. A manual inspection of errors revealed that a significant number (about 20%) of them are due to user-specific procedures that were taught to the agent by human users during the original study. These errors are too hard to resolve with a batch training approach, and would require incorporating user-specific behavior in the semantic parsing.

6.6 Discussion

In this chapter, we presented a new approach for training semantic parsers, which incorporates conversational context. We developed a structured prediction formulation that incorporates conversational context by leveraging structural regularities in conversation sequences. This enables joint learning of text-based features traditionally used by seman-

tic parsers, as well as structural features to model the flow of discourse. While the current work uses a simple model of discourse as statistical regularities (with Markov properties) in sequential structure of conversations, this can further be refined by incorporating models of discourse entities and discourse referents from discourse representation theory. Understanding of conversations can also be enhanced by models incorporating background world knowledge. Finally, the idea of using conversational context can be generalized to incorporate other modes of contextual information, such as from the agent's execution environment. Our contributions are:

- An annotated dataset of conversations, comprising of 4759 natural language statements with their associated logical forms.
- A latent variable approach that incorporates both text-based cues within sentences (that model semantics), and structural inferences across sentences (that capture discourse and pragmatics). Using this, we empirically demonstrate significant improvements in parsing conversations over the state-of-the-art.
- Latent categories learned by the approach are seen to be semantically meaningful, and can be interpreted in terms of discourse states in the conversations.

Chapter 7

Learning Question-asking Strategies for Mixed-Initiative Dialog

When children learn from language, they do not rely only on passively receiving supervision from a teacher in the form of explanations or instructions. Rather, the interaction usually takes the form of a mixed-initiative dialog, where students can also ask questions and proactively seek clarifications that simplify the learning problem. These questions may focus on generalizing learning to novel situations, exploring new hypotheses, or filling crucial information gaps. The ability to ask questions can, therefore, fundamentally facilitate learning and understanding. The central value of questions (in terms of human pedagogy) is indicated by the fact that a student's understanding of a topic is often judged from being able to ask the right questions.

Language allows a natural medium for such interactive dialog on part of a learner. In this chapter, we explore the ability to ask questions in an interactive dialog-based setting. We examine the role of mixed-initiative interaction between a learner and a teacher in terms of the following question: how can we learn intelligent questioning strategies on part of the learner, which can help concept learning tasks? Interactivity on the part of the learner can facilitate multiple facets of learning. While the potential space of questions

that a student can ask a teacher can be vast (Ram, 1991), our treatment here specifically focuses on the ability of interactive dialog for:

1. Seeking labels for specific examples
2. Asking for explanations of a concept
3. Requesting clarifications about earlier explanations

These dimensions can facilitate multiple aspects of the learning process: including learning from labeled examples (similar to traditional supervised learning), learning from natural language explanations (extending our research presented in the previous chapters) and alleviating limitations in the learner's semantic parsing abilities. Learning systems that reify these abilities can enable users to teach new concepts using a blend of traditional and natural language supervision in dialog-based settings.

In this chapter, we first describe how our approach to learn from declarative language described in Chapter 4 can be extended to learn classifiers using a mix of explanations and labeled examples. Next, we present a simple reinforcement learning approach for learning question-asking policies for dialog based concept learning.

7.1 Related Work

From the perspective of traditional supervised learning, the problem of asking questions can be seen as cognate with active learning. Methods in active learning have explored various criteria for choosing which of a set of unlabeled examples to label next while training supervised machine learning models. This can be seen as asking a specific kind of question – and also the most natural – since labeled examples compose the supervision in traditional machine learning models. Thus, learning to ask questions subsumes active learning, but also generalizes it in multiple ways.

A learner can initiate dialog to solicit various kinds of measurements to simplify a

learning problem. These include seeking of instance labels (e.g., 'This email is spam'), feature labels ('Emails that mention "Viagara" are spam'), label proportions ('Spam emails are as common as non-spam email'), constraints on model expectations, etc. However, using natural language as a medium of information communication adds further challenges: the optimal question answering strategy may depend not only on the learning task at hand, but also the teacher's skill and the quality of the learner's semantic parsing model. For example, a teacher's language may be too complex to handle for the parser, in which case it might be preferable to stick to asking about the teacher about instance labels (which would not require parsing).

However, rather than learning static criterion for choosing what question to ask (such as in active learning), our focus here is to learn to ask questions in context of a conversational setting (which is inherently a dynamic process). To explain, asking a teacher to rephrase an explanation only makes sense in specific contexts (when the interpretation of something said previously is unclear). This has motivated our choice of a reinforcement learning based approach for learning question-asking strategies rather than static criterion.

Conversational agents and dialog systems have been explored in comprehensive details in the field of dialog research. However, our focus here is not the mechanism of language generation, but learning what questions to ask at specific points in the concept learning process. To simplify our analysis, we restrict ourselves to predefined questions types mentioned above.

7.2 Approach

Learning question-asking strategies presupposes a mechanism through which answers to those questions can jointly help towards the goal of the dialog, i.e. concept learning. Thus, in Section 7.2.1 we first describe an approach for learning classifiers from a mix of labeled examples and natural language advice as a preliminary. Next, in Section 7.2.2, we

present our reinforcement learning formulation for learning question-asking strategies in a simulated conversational setting. The space of actions for the framework consists of a vocabulary of question types that a learner can ask a teacher, and reward is evaluated in terms of improvements in the concept model that an asked question leads to.

7.2.1 Learning classifiers from a mix of observations and explanations

We can naturally extend our approach presented in Chapter 4, to learning loglinear classifiers from a mix of both labeled and unlabeled data, and natural language explanations. This is done by simply appending a log-likelihood term for the labeled examples to the objective in Equation 4.1. The updated objective is given by:

$$J_Q(\theta) = \mathcal{L}_{labeled}(\theta) + \mu\Big(\mathcal{L}_{unlabeled}(\theta) - \min_{q \in Q} KL(q \mid p_\theta(Y|X))\Big) \tag{7.1}$$

Here, $\mathcal{L}_{labeled}(\theta) = \sum_k l_\theta(x_k, y_k)$ denotes the log-likelihood term for a set of $n_{labeled}$ labeled examples $\mathcal{X}_{labeled} = \{(x_k, y_k)\}_{labeled}^n$, whereas the other two terms are as before: $\mathcal{L}_{unlabeled}(\theta)$ denoting log-likelihood over a set of $n_{unlabeled}$ unlabeled examples, and a posterior regularizer term (KL-divergence) penalizing violations of the parse natural language advice. The training procedure proceeds similar to before. In the E-step, the computation of the posterior regularizer remains unchanged. However, the M-step is modified so that the classifier parameters θ are learned using both the inferred labels (from the E-step) for the unlabeled examples, and provided labels for the labeled examples.

In Equation 7.1, $\mu > 0$ determines the relative weights of provided example labels and natural language advice in the optimization objective. In cases where there is little labeled data, we would like to rely primarily on constraints specified from natural language explanations, and unlabeled data. On the other hand, in scenarios where there is a lot of labeled data available enabling robust inductive inference, we would like to primarily rely

on it rather than explanations[1]. While setting up the optimization problem, the value of μ can be adapted to reflect this intuition. In our experiments, we found setting $\mu = 1/n_{labeled}$ to work well across settings.

7.2.2 RL formulation for learning to question

For our framework for concept learning in a dialog setting, we assume the presence of a teacher to answer questions posed by the learner. We further restrict the structure of dialog to the learner asking a sequence of questions, and the teacher responding to them. The schematic in Figure 7.1 shows a hypothetical example of such a dialog.

Further, we also assume the presence of a held-out set of labeled examples of the concept, which can be used to evaluate the classification performance of the learner's concept model as the dialog progresses. At each step t, the learner's action a_t consists of choosing a question to ask the teacher. The teacher's response to the learner's question is parsed to a data measurement (in the form of a labeled example, or a model constraint), which is then incorporated into the learner's concept model (by retraining the model with the additional labeled example or the new quantitative constraint). The classification performance, c_t, of the updated model is evaluated on the held-out set. The change in classification performance from the previous step, r_t approximates the marginal value of the question in the learning process, and constitutes the learner's immediate reward at that step.

Our approach for learning question asking strategies consists of modeling the dialog process as a Markov Decision Process, and using a SARSA-learning procedure to estimate the relative values of different question types in different contexts.

We next describe the state-space, actions and rewards, and the learning procedure.

[1]To explain, in the asymptotic case of infinite data with labels, an inductively learned Bayes Classifier would be optimal

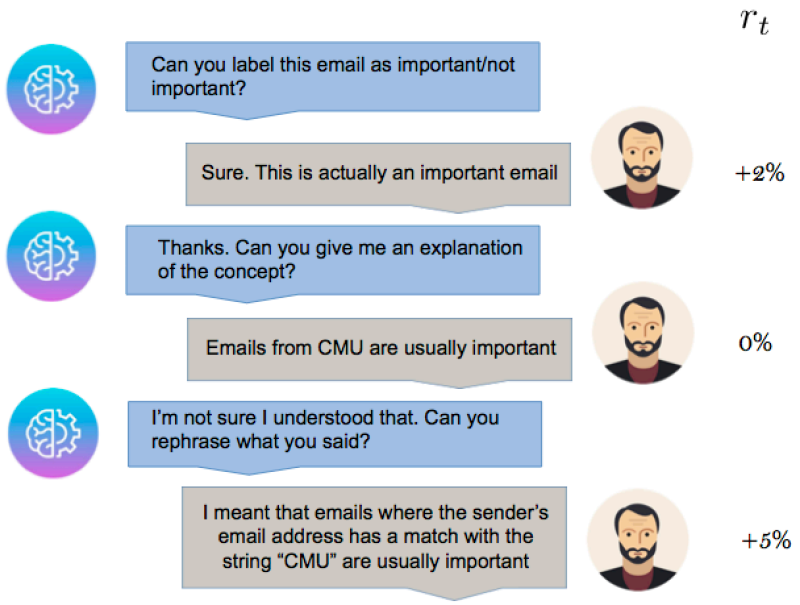


Figure 7.1: We assume that the dialog between the learner and the teacher is in the form of turn-wise conversations – consisting of a sequence of questions asked by the learner, and the teacher's responses to those questions. At each step in this process, the teacher's response is parsed by the learner, and can be incorporated into the learner's concept model as either a labeled example or a quantitative constraint (the learner can also choose to seek a clarification). A reward (denoted by r_t) can be computed at each step, which denotes the marginal change in classification performance on a held-out set of examples due to the last response. In this framework, learning good question-asking strategies corresponds to asking sequences of questions that maximize the cumulative (discounted) reward, and hence quickly lead to effective concept models

Action Space

As mentioned previously, there can be a multitude of questions that a human learner can ask a teacher in context of concept learning tasks. However, in this chapter, we are concerned only with three specific types of questions that are specially germane for facilitating learning from a mix of labeled examples and explanations. These consist of the following:

1. **Seeking labels for specific examples:** This is similar to traditional active learning. In particular, we can have a different action corresponding to every active learning criterion, which chooses which example to label next. In our experiments, we use two active learning techniques (and hence have a corresponding action for each):

    - *Random Instance:* Ask for the concept label for a randomly chosen unlabeled instance in the data
    - *Maximum Uncertainty Instance:* Ask for the concept label for the instance in the data for which the current concept model is most uncertain (highest entropy). If there are multiple such instances, randomly pick one of them.

2. **Asking for an explanation for the concept:** This action seeks out from the teacher a short natural language explanation of the concept, similar to those seen in previous chapters. These are intended to be incorporated in the concept model as quantitative constraints. In general, this can encompass several types of explanations. For example,

    - Asking for probability estimates about specific labels and features. e.g., 'How likely are emails about meetings?'
    - Asking for discriminative features for particular concept labels. e.g., 'Can you think of a feature that if present always denotes that an email is important?'

- Asking about class probabilities. e.g., 'Around what fraction of emails in your inbox are important?'

All of these provide admissible constraints which our classifier training procedure (from Section 7.2.1) can handle. However, to simplify our analysis and data collection, in our experiments, we only explored asking for general explanations of the form '*Can you give me an explanation of the concept?*', which could return a wide variety of responses.

3. **Requesting clarifications about earlier explanations:** This action asks for a clarification about the interpretation of a previous explanation (which should be helpful in cases when the learner is uncertain about the correct interpretation). Such a clarification could simplest verify if the learner's interpretation of the previous explanation (parsed logical form) was correct or not. This type of binary feedback about language interpretation has been previously explored in works such as (Goldwasser and Roth, 2014) to train semantic parsers. In our case, this feedback is simply used to decide whether the previous explanation should be used to update the learner's concept model, based on whether it was interpreted correctly.

Rewards

The reward, r_t, evaluates the change in concept learning performance due to an asked question at each step of the dialog. In our experiments, we use the model's F1 score as the measure of concept learning performance, c_t. We define the reward as the absolute change in model performance from the previous step, $r_t = c_t - c_{t-1}$.

State-space

Conversations are inherently dynamic. The best question to ask at a particular point might strongly depend on the state of the conversation, i.e. this could include the pedagogical

stage in the learning process (exploratory vs confirmatory), previous questions asked, etc. Thus, defining a rich enough state space is an important consideration for a formulation of conversational learning.

In our treatment, we assume a discrete state-space, which is defined simply by the cross product of the following (also discrete) factors.

- Curricular stage in the learning process: We use a discrete variable to model the curricular stage of the learner simply by the number of steps (questions previously asked) in the interaction at any point. We cluster the number of steps such that the curricular stage are defined by the following five values: BEGINNER (0 steps), ELEMENTARY (1-5 steps), INTERMEDIATE (6-10 steps), ADVANCED (11-15 steps) and MATURE ($>$ 15 steps)[2].
- Reward in the previous state: We discretize the value of reward using the following threshold values: GOOD ($r_t \geq 0.03$), INCREASING ($0.01 < r_t < 0.03$), UNCHANGED ($-0.01 < r_t < 0.01$) and DECREASING ($r_t < -0.01$).
- Velocity of reward (3 states): This simply returns whether the value of the discrete variable for the reward (just described) improves, worsens or remains the same compared to the previous step.
- Type of the previous two actions as earlier mentioned in the description of the Action Space.
- Domain of learning task: This simply indicates which domain of classification tasks the current concept belongs to. The evaluation of our approach uses datasets described in previous chapters, corresponding to three types of tasks: Email categorization, Shape classification and Bird species identification; hence this variable can take three corresponding values.

[2]These threshold values were heuristically chosen based on the observation that for most of datasets that we experiment with, classifier performance roughly begins to plateau at around 20 training examples

- Confidence of previous parse: We model the confidence of the parser in parsing the previous sentence as the ratio of the probability of the highest probability (predicted) logical form and the next best logical form. We discretize this ratio into three values, corresponding to the upper, lower and interquartile ranges for the value of the ratio.

On the other hand, we note here that our state-space does not model the following also important considerations:

- User behavior: We do not model aspects such as whether the user is a good teacher (i) provides correct information, (ii) uses parseable language
- Task difficulty: how expressible is the task using language explanations. For example, some concept maybe significantly easier to explain using language than others, depending on the logical language available for grounding explanations. For example, it maybe impossible to explain digit recognition using pixel level features using natural language (such as in the MNIST classification dataset).

7.2.3 Model training

For training, we use the SARSA learning algorithm (Rummery and Niranjan, 1994), where we represent the Q-value function for a pairs of a state s and an action a, $Q(s, a)$, as a table of values. We use an ϵ-greedy strategy (where we set $\epsilon = 0.10$). In other words, the strategy balances between exploitation and exploration by picking the next action to be the estimated optimal one (in terms of having the maximum estimated Q-value for a particular state) with a probability of $1 - \epsilon$, and choosing the next action randomly with a probability of ϵ.

SARSA allows on-policy learning by iteratively updating the estimates of $Q(s, a)$ values at each step. In each step t, when the agent is in state s_t, it chooses the next action a_t ϵ-greedily, and observes the corresponding reward r_{t+1} and the next state s_{t+1}. The agent again chooses the next action a_{t+1} ϵ-greedily, and makes the following update.

$$Q(s_t, a_t) \longleftarrow Q(s_t, a_t) + \alpha\big(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\big) \tag{7.2}$$

Here, $\gamma \in (0, 1]$ denotes the discounting factor for the rewards (set to 0.9 in our experiments), while $\alpha \in (0, 1]$ (set to 0.2 in our experiments) denotes the learning rate

7.3 Evaluation

For our model training, our environment does not consist of questioning human users in real-time, but we rather simulate the conversational exchange by asking questions to a oracle, which has access to previously collected data about a concept (consisting of both explanations and labeled examples). In particular, we note that each of the question types that we previously discussed in Section 7.2.2 – namely asking for instance labels, asking for a general concept explanation, verifying interpretation of an explanation, or asking for a paraphrase – can be answered with statically collected data – consisting of labeled examples and natural language explanations paired with their logical forms and paraphrases.

For our experiments, we use data previously described in Chapter 3 and Chapter 4, consisting of three domains of classification tasks – Email concepts, Bird Species identification and Visual Shape classification. We compared our model with a naive policy that randomly takes a new action at each step in the learning process.

We note that a limitation of learning question-asking strategies from simulated interactions is that for some classification tasks, we may run out of concept explanations in the course of model training (since the number of explanations of a concept are limited). In such cases, we end the interaction as soon as all explanations are already provided to the learner.

7.3.1 Learned vs Random policies

Figure 7.2 shows cumulative reward (averaged over 20 classification tasks) in the process of interactive concept learning when the question asking strategy is learned compared with the random strategy. Here, the question-asking strategy was previously learned over a set of 50 separate classification tasks. We note that the policy learning strategy performance is consistently superior (on average, it achieves any given level of classification performance in a smaller number of interactions), which unambiguously indicates value in asking the right questions.

Figure 7.2: Cumulative Reward (averaged over 20 tasks) for interactive concept learning when question asking strategy is learned vs when questions are asked randomly

7.3.2 Reliance on NL vs Parsing accuracy

Intuitively, the ability of a learner to interpret language should be a significant consideration in whether it should choose to ask for explanations or labeled examples. To test this, we simulate scenarios of learners with different levels of semantic parsing ability by choosing the true logical form for any explanation with the corresponding probability, and choosing an alternative logical form from the remaining candidates in the beam otherwise.

Figure 7.3 depicts the effect of parsing competence on learned question-asking strate-

gies. We note that the learned strategies increasingly avoid seeking natural language explanations of concepts as the parsing performance worsens. In the base case where the learner has no parsing competence, the model learns to exclusively ask for labeled examples only (In the figure, the fraction is seen to converge to 0.1 instead of 0.0 due to the ϵ-greedy nature of the policy-learning procedure).

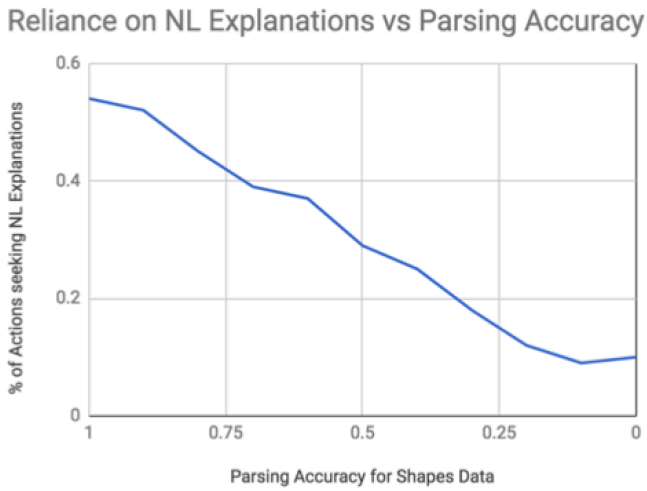


Figure 7.3: Fraction of actions seeking Natural Language Explanations vs competence of the learner's semantic parsing model

7.3.3 Differential value in sequences of explanations

A significant feature in concept learning from explanations, which our approaches do not acknowledge is that a person's multiple explanations of a concept might have significantly different utility. Specifically, it might be reasonable to imagine that people would be more likely to provide the most useful explanations first, and provide minor explanations subsequently.

From an ablation study, we observe that this is indeed a valid concern. Figure 7.4 shows the average marginal increase in classification performance over 50 visual shape classification tasks from explanations with different rank (based on order of providing from a human user).

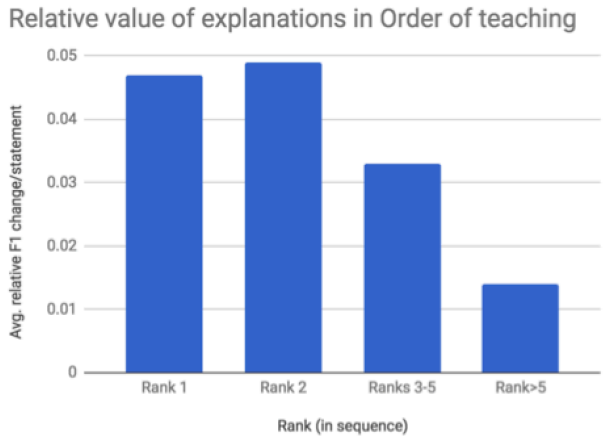


Figure 7.4: Average increase in classification performance by incorporating explanations of different positional orders in explanation sequences

7.4 Discussion

Learning to ask questions is a fundamentally important aspect of human learning. In this chapter, we provided a simple formulation for learning to ask different types of questions in the context of interactive concept learning. Our empirical analysis shows that question-asking strategies learned from our approach can accelerate concept acquisition. At the same time, a direction that we don't explore here is that the interactive nature of the dialog naturally lends itself to simultaneously improving the learner's semantic parsing capabilities. Our approach of grounding language in teacher-learner interactions with a shared and measurable goal (concept acquisition) can have interesting implications beyond concept learning; as a modeling framework, it is also consonant with the cooperative principle of language (Grice, 1975).

On the other hand, our approach is simplistic in some ways. Significantly, while our method can consume both explanations and labels, we make restrictive assumptions on the types of questions that can be asked, as well as the structure of the dialog. In particular, for the sake of simplicity, our framework does not allow more than two exchanges about a particular explanation – consisting of the original explanation from the teacher, and at most one clarification.

Further, in future work, it might be possible to also evaluate the learned question-asking strategies through human user studies.

Chapter 8

Conclusion

In this research, we have explored how different aspects of natural language can be leveraged to train classification models from language interactions with human users. As our results across multiple tasks show, such interactions can be effective sources of supervision for training statistical models, rivaling the performance of models trained from fully labeled data in many cases. Our approach for learning from natural language rests on semantic parsing, and can be seen as a fundamentally new application of semantic parsing in comparison with prior work. A second, but important component of our work is the development of new methods for grounded language acquisition that use context from the environment to improve the process of semantic interpretation. Leveraging the grounding of language to improve both language understanding and learning about the environment is a central theme in our work.

8.1 Overview

We began by exploring the problem of how natural language explanations can be used to specify relevant features for concept learning tasks. Through empirical studies on the domain of emails, we showed that parsing such explanations to feature functions can lead

to learning with fewer examples than traditional classification methods for email classification. Note that language can not only highlight specific features of the data from among a pre-defined set (which is more related to the problem of *feature selection*), but also compositionally define new features, which can lead to representations that make inductive learning easier. An intrinsic property of natural language that facilitates this is that the structure of language explanations closely mirrors the structure of the computation needed to calculate a feature. For example, a person explaining the concept of an email that is a reminder to oneself might say *'The sender and the recipient address are the same'*.

We then explored how didactic language can be used to directly provide supervision for training of concept learning models, reducing (or even eliminating) dependence on labeled data. Here, we introduce a general quantitative framework through which a set of explanations are converted to quantitative constraints, which can be used to learn classifiers by inference over a set of unlabeled examples of the concept. Our formulation defines a modified optimization problem for model training, where the objective is to find parameter estimates for the classifier that do not simply fit the data, but also agree with the human provided natural language advice as much as possible. In particular, we focus on identifying different types of statistical relationships between features and labels expressed in language (e.g., 'If the subject of an email threatens to close an account, it is definitely spam') and leverage linguistic quantifiers (e.g., 'definitely'). Our experiments on multiple data domains suggest that this approach for learning driven by linguistic quantification is surprisingly effective.

In the next two chapters, we focused on the complementary problem of learning to interpret language utterances. In particular, we provided two new algorithms for learning semantic parsers that incorporate different types of situational context in the process of language interpretation. First, we considered the problem of jointly learning both a semantic parser and a concept classifier from natural language explanations, and a small number of labeled examples of the concept. Our approach leverages pragmatics to prefer

interpretations of natural language explanations that are more discriminative in context of concept learning. Our empirical results show that sensory context (in the form of feature value observations) along with natural language instruction and labeled examples can be a viable and inexpensive source of supervision for training semantic parsers. This is a significantly weaker form of supervision than has been previously used for training semantic parsers. We believe that the approach of modeling *latent interpretations of language* as auxiliaries to downstream tasks, which we propound here, can be a useful paradigm for grounded language learning in a range of applications.

Next, we presented a semantic parsing method for parsing conversations. Our approach incorporates conversational context by leveraging structural regularities in the sequential structure of conversations. The method employs a structured prediction formulation, enabling it to incorporate not only text-based features traditionally used by semantic parsers, but also structural features to model the flow of discourse. For example, a terse statement such as 'Use mom's email' can be better parsed using such discourse features that can model the intent of the speaker from previous statements, such as 'Compose a new email'.

Finally, we explore the problem of learning intelligent question-asking strategies to engage with a teacher, which can help concept learning tasks in interactive dialog settings. Here, we developed a simple reinforcement learning formulation for the problem, where agent actions correspond to different question types and reward is measured in terms of improvement in performance on concept learning task. Our empirical analysis across three domains of classification taks shows that question-asking strategies learned from our approach can accelerate concept acquisition.

8.2 Summary of contributions

The principal contributions of this work can be summarized as follows:

1. We present a novel application of semantic parsing for interactive feature specification

for machine learning tasks.

2. We present the first application of linguistic quantifiers for guiding generalization of statistical models. Our approach emulates paradigms such as constraint-driven learning that leverage declarative knowledge in lieu of example labels for supervision – but also extends them by providing a natural solution to how such constraints are to be obtained.

3. We develop datasets for concept learning from natural language explanations for three data domains. These include classification tasks with various levels of difficulty.

4. We provide two new approaches (and corresponding datasets) for semantic parsing in grounded contexts. Our approach in Chapter 5 represents a significant departure from existing methods in semantic parsing by learning from weaker supervision than annotations of logical forms or their denotations for individual sentences.

5. We present a framework for learning from a blend of labeled examples and natural language explanations, which might be the most realistic scenario for concept-learning tasks in everyday life. Using this framework, we explore the problem of learning questioning strategies for mixed initiative dialog, and develop a reinforcement learning formulation for this.

8.3 Directions for future work

Many aspects of this work have an exploratory nature. A reasonable characterization of the research done in this dissertation might be as a preliminary study in leveraging advances in language technologies and the newfound ubiquity of sensor-effector systems grounded in the environment towards the goal of creating machine learning systems that can be taught through language, similar to a how one might teach a human assistant. Our exploration of learning from language revealed several challenges and directions for future work.

8.3.1 Learning from multiple teachers

An important challenge is developing algorithms that can learn from multiple teachers. e.g., consider an agent trying to learn to identify people who are at risk of a specific type of cancer by reading advice from the web. In such scenarios, the issue of trustworthiness may be an important modeling consideration, since some teachers may be more reliable than others. One way to accommodate this might be through extending our approach in Chapter 4 by weighing constraints from different sources differently.

8.3.2 Learning persistent knowledge and ontologies

Another important aspect that our treatment here neglects is learning from extended interactions. When humans learn concepts, they do not learn each concept in isolation. Rather, they might rely on using knowledge of simpler concepts to learn incrementally more complex concepts. Further, in many cases, jointly learning concepts might be an easier learning problem than learning concepts in isolation due to coupling constraints. For example, an explanation such as 'Emails from my friends are definitely important' couples two concepts: emails that are important, and contacts who are friends. Learning one concept should help learn the other. In future work, the research presented here can be expanded to enable learning agents with persistent knowledge bases, which allow learned concepts and abstractions to be reused across multiple tasks – enabling generalized learning. As an example, the concept of a 'diagonal' acquired while learning the game of tic-tac-can be re-used in learning other games (such as chess).

8.3.3 Characterizing what to learn from language

An important direction of future research that we do not explore here is the question of choosing which learning tasks might be better to learn through language. The renaissance of neural network methods in the current decade has led to exciting advances in supervised

learning for tasks such as object recognition, which seem closer to low-level perception than higher-level reasoning. This type of learning may be similar to things that humans learn in an unconscious fashion (e.g., learn to see, learn to walk, etc.). On the other hand, when people learn to play a game (e.g., tic-tac-toe), it is clear that they use a different paradigm, where good strategies and sub-goals can be explained, and learning does not need to be driven solely by examples. This kind of learning can enable better generalization and reasoning across tasks, compared to fixed-structure learning (For example, many deep network video game players can be thrown off by simply mapping pixels to different color values). In general, it is unclear how neural models can be used in the long tail of learning tasks that have limited data, or require complex reasoning. Along with advances in speech and language processing, this suggests a future towards systems that can leverage the perceptual capabilities of neural methods with the explainability of language.

8.3.4 Natural language programming

One of the exciting possibilities in learning from language is to program computers using natural language. The ubiquity of computing devices with rich repertoires of sensor-effector capabilities can have vast potential if language can be used to unfetter the creativity of everyday people who do not know programming languages. By enabling agents that can be instructed through language, people can program a vast variety of personalized procedures and behavior.

The work presented in this dissertation propounds the idea of using language to guide learning of statistical models. This is an intriguing direction, which contrasts starkly with the predominant theme of using statistical learning methods to advance the field of NLP. We believe that language may have as much to help learning, as statistical learning has helped NLP. Machines that can be interactively instructed from natural language present

a profoundly exciting new area for the imminent future, which can have transformational implications for both learning and language research.

Appendix

Appendix A: List of common notations

x	Input instance from an input space $\mathcal{X}$
y	Output label from an output space $\mathcal{Y}$
s	A natural language statement
l	Logical Form for natural language statement
$\mathcal{G}(s)$	Semantic parsing grammar; takes as input a statement s and returns a set of candidate logical forms that can be associated with s
$\psi(s, l)$	Feature function for semantic parsing; takes as input a statement s and candidate logical form l, and returns a real-valued feature vector
θ_p	Model parameters for a semantic parser
θ_c	Model parameters for a concept classifier
$[\![l]\!]_x$	Evaluation of a logical form l in context of a data instance x; equivalently, the denotation of l in context x
$\phi(x, y)/\phi(X, Y)$	Feature function defined on an instance x and an output label y, or a set of instances $X = \{x_i, x_2 \dots x_n\}$ and a set of output labels $Y = \{y_1, y_2 \dots y_n\}$. Returns a real valued feature value
$\otimes$	Cartesian product of sets
$\mathbb{R}^n$	Euclidean space with dimensionality n
$u^T v$	Dot product of two vectors $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^n$, equivalent to $\sum_i^n u_i v_i$

Appendix B: Relevant publications

- (Srivastava et al., 2018) Zero shot learning of classifiers from natural language quantification. Proceedings of ACL 2018
- (Labutov et al., 2018) LIA: A natural language programmable personal assistant

(Demonstrations). Proceedings of EMNLP 2018

- (Srivastava et al., 2017c) Learning classifiers from declarative language. Workshop on Learning from Limited Data, NIPS 2017
- (Srivastava et al., 2017b) Joint concept learning and semantic parsing from natural language explanations. Proceedings of EMNLP 2017
- (Srivastava et al., 2017a) Parsing natural language conversations with contextual cues. IJCAI 2017

In preparation/review

- Learning question-asking strategies for mixed-initiative dialog. In preparation
- An agent for learning new natural language commands. In review

Bibliography

Jacob Andreas, Dan Klein, and Sergey Levine. Learning with latent language. *CoRR*, abs/1711.00482, 2017. URL `http://arxiv.org/abs/1711.00482`. 4.1

Aristotle. Categoriae, 4th Century B.C. 3

Yoav Artzi and Luke Zettlemoyer. Bootstrapping semantic parsers from conversations. In *Proceedings of the conference on empirical methods in natural language processing*, pages 421–432. Association for Computational Linguistics, 2011. 6.1

Yoav Artzi and Luke Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1(1):49–62, 2013. 2.2, 6.3.1

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. Broad-coverage ccg semantic parsing with amr. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1198`. 2.2, 3.2.1

Amos Azaria, Jayant Krishnamurthy, and Tom M Mitchell. Instructable intelligent personal agent. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2681–2689. AAAI Press, 2016. 2.1, 6.3.1, 6.4, 6.5.1

Elke Bach, Eloise Jelinek, Angelika Kratzer, and Barbara BH Partee. *Quantification in natural languages*, volume 54. Springer Science & Business Media, 2013. 4.1

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 6.5.1

Jon Barwise and Robin Cooper. Generalized quantifiers and natural language. *Linguistics and philosophy*, 4(2):159–219, 1981. 4.1

Elizabeth Bates. *Language and context : the acquisition of pragmatics*. Academic Press New York, 1976. ISBN 0120815516 0120815508. 2.2, 6.1

Kedar Bellare, Gregory Druck, and Andrew McCallum. Alternating projections for learning with expectation constraints. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 43–50. AUAI Press, 2009. 4.2.2

Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1415–1425, 2014. URL `http://aclweb.org/anthology/P/P14/P14-1133.pdf`. 6.1

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, 2013. 2.2, 3, 6.1, 6.2.2

Alan W Biermann. Natural language programming. In *Computer Program Synthesis Methodologies*, pages 335–368. Springer, 1983. 2.1

A. Blum. Learning boolean functions in an infinite attribute space. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 64–72, New York, NY, USA, 1990. ACM. ISBN 0-89791-361-2. doi: 10.1145/100216.100224. URL `http://doi.acm.org/10.1145/100216.100224`. 3.1

Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997. 3

SRK Branavan, Harr Chen, Luke S Zettlemoyer, and Regina Barzilay. Reinforcement learning for mapping instructions to actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL: Volume 1-Volume 1*, pages 82–90. Association for Computational Linguistics, 2009. 2.1

SRK Branavan, David Silver, and Regina Barzilay. Learning to win by reading manuals in a monte-carlo framework. *Journal of Artificial Intelligence Research*, 43:661–704, 2012. 2.1, 4.1

Nathanael Chambers and Daniel Jurafsky. Unsupervised learning of narrative event chains. In *ACL*, volume 94305, pages 789–797, 2008. 6.1

Ming-Wei Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 280–287, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P07-1036`. 2.1, 4.1, 4.4

Noam Chomsky. *Language and mind*. Cambridge University Press, 2006. 1

James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. Driving semantic parsing from the world's response. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 18–27. Association for Computational Linguistics, 2010. 2.1, 6.1

Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics, 2002. 3.2.1, 6.2.1

Gregory Druck, Gideon Mann, and Andrew McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international*

ACM SIGIR conference on Research and development in information retrieval, pages 595–602. ACM, 2008. 4.4.1

Robin Dunbar. *Grooming, gossip, and the evolution of language*. Harvard University Press, 1998. 1

Jacob Eisenstein, James Clarke, Dan Goldwasser, and Dan Roth. Reading to learn: Constructing features from semantic abstracts. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 958–967. Association for Computational Linguistics, 2009. 2.1

Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013. 4.1

John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957. 1.5.1

Michael Fleischman and Deb Roy. Intentional context in situated natural language learning. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 104–111. Association for Computational Linguistics, 2005. 2.2

New Dataset For, Bryan Klimt, and Yiming Yang. The enron corpus:. In *In ECML*, pages 217–226, 2004. 3.3

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. URL `http://dl.acm.org/citation.cfm?id=1625275.1625535`. 3.4.1

Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–

2049, 2010. (document), 2.1, 4, 4.1, 4.3, 4.2.2

Dan Goldwasser and Dan Roth. Learning from natural instructions. *Mach. Learn.*, 94 (2):205–232, February 2014. ISSN 0885-6125. doi: 10.1007/s10994-013-5407-y. URL `http://dx.doi.org/10.1007/s10994-013-5407-y`. 2.1, 6.1, 3

Alison Gopnik, Clark Glymour, David M Sobel, Laura E Schulz, Tamar Kushnir, and David Danks. A theory of causal learning in children: causal maps and bayes nets. *Psychological review*, 111(1):3, 2004. 1

H Paul Grice. Logic and conversation. *1975*, pages 41–58, 1975. 1, 7.4

Sumit Gulwani and Mark Marron. Nlyze: Interactive programming by natural language for spreadsheet data analysis and manipulation. In *SIGMOD*, 2014. 3

Michael AK Halliday. Towards a language-based theory of learning. *Linguistics and education*, 5(2):93–116, 1993. 2.1

David Harel, Assaf Marron, and Gera Weiss. Behavioral programming. *Commun. ACM*, 55(7):90–100, July 2012. ISSN 0001-0782. doi: 10.1145/2209249.2209270. URL `http://doi.acm.org/10.1145/2209249.2209270`. 2.1

Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42 (1-3):335–346, 1990. 1.5.1

Ryuichiro Higashinaka, Mikio Nakano, and Kiyoaki Aikawa. Corpus-based discourse understanding in spoken dialogue systems. In *ACL*, pages 240–247, 2003. 6.1

Scott B Huffman and John E Laird. Learning procedures from interactive natural language instructions. In *Proc. 10th International Conference on Machine Learning. Amherst, MA*, pages 143–150, 2014. 2.1

Siddharth Karamcheti, Edward C Williams, Dilip Arumugam, Mina Rhee, Nakul Gopalan, Lawson LS Wong, and Stefanie Tellex. A tale of two draggns: A hybrid approach for interpreting action-oriented and goal-oriented instructions. *arXiv preprint*

arXiv:1707.08668, 2017. 4.1

Rohit J Kate, Yuk Wah Wong, and Raymond J Mooney. Learning to transform natural to formal languages. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1062, 2005. 3

Angelika Kimmig, Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. A short introduction to probabilistic soft logic. In *NIPS Workshop on Probabilistic Programming: Foundations and Applications*, 2012. 4.1

Parisa Kordjamshidi, Dan Roth, and Kristian Kersting. Systems ai: A declarative learning based programming perspective. In *IJCAI*, pages 5464–5471, 2018. 2.1

Samantha Krening, Brent Harrison, Karen M Feigh, Charles Lee Isbell, Mark Riedl, and Andrea Thomaz. Learning from explanations using sentiment and advice in rl. *IEEE Transactions on Cognitive and Developmental Systems*, 9(1):44–55, 2017. 2.1

Jayant Krishnamurthy. Probabilistic models for learning a semantic parser lexicon. In *NAACL-HLT*, pages 606–616, 2016. 3.2.1, 6.5.1

Jayant Krishnamurthy and Thomas Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of Association for Computational Linguistics*, 2013. 6.1

Jayant Krishnamurthy and Tom M Mitchell. Weakly supervised training of semantic parsers. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 754–765. Association for Computational Linguistics, 2012. 2.2, 5.1.2, 6.1

Howard S Kurtzman and Maryellen C MacDonald. Resolution of quantifier scope ambiguities. *Cognition*, 48(3):243–279, 1993. 4.1

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Inducing probabilistic ccg grammars from logical form with higher-order unification.

In *EMNLP*, pages 1223–1233, 2010. URL http://dl.acm.org/citation.cfm?id=1870658.1870777. 6.1

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke S Zettlemoyer. Scaling semantic parsers with on-the-fly ontology matching. In *ACL*, 2013. 2.2, 6.2.2

Igor Labutov, Shashank Srivastava, and Tom Mitchell. LIA: A natural language programmable personal assistant. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2018. 1, 8.3.4

John E Laird. *The Soar cognitive architecture*. MIT press, 2012. 2.1

John E Laird, Kevin Gluck, John Anderson, Kenneth D Forbus, Odest Chadwicke Jenkins, Christian Lebiere, Dario Salvucci, Matthias Scheutz, Andrea Thomaz, Greg Trafton, et al. Interactive task learning. *IEEE Intelligent Systems*, 32(4):6–21, 2017. 2.1

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 3.1, 4.1

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014. 4.1

Staffan Larsson and David R. Traum. Information state and dialogue management in the trindi dialogue move engine toolkit. *Nat. Lang. Eng.*, pages 323–340, 2000. ISSN 1351-3249. doi: 10.1017/S1351324900002539. URL http://dx.doi.org/10.1017/S1351324900002539. 6.1

Tessa A. Lau, Clemens Drews, and Jeffrey Nichols. Interpreting written how-to instructions. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1433–1438, 2009. URL

http://ijcai.org/Proceedings/09/Papers/240.pdf. 2.1

Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196, 2014. URL http://jmlr.org/proceedings/papers/v32/le14.html. 3.4.1

Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 4.1

Jay L Lemke. *Talking science: Language, learning, and values.* ERIC, 1990. 2.1

Percy Liang. Lambda dependency-based compositional semantics. *CoRR*, abs/1309.4408, 2013. URL http://arxiv.org/abs/1309.4408. 2.2

Percy Liang and Christopher Potts. Bringing machine learning and compositional semantics together. *Annu. Rev. Linguist.*, 1(1):355–376, 2015. 5.1.2

Percy Liang, Michael I Jordan, and Dan Klein. Learning from measurements in exponential families. In *Proceedings of the 26th annual international conference on machine learning*, pages 641–648. ACM, 2009a. 4.1

Percy Liang, Michael I Jordan, and Dan Klein. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 91–99. Association for Computational Linguistics, 2009b. 2.1, 2.2

Percy Liang, Michael I Jordan, and Dan Klein. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 590–599. Association for Computational Linguistics, 2011. 2.2, 4.2.1, 5.1.2

Percy Liang, Michael I Jordan, and Dan Klein. Learning dependency-based compositional semantics. *Computational Linguistics*, 39:389–446, 2013. 6.2.1

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`. 4.2.1

Huan Ling and Sanja Fidler. Teaching machines to describe images with natural language feedback. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5075–5085, 2017. URL `http://papers.nips.cc/paper/7092-teaching-machines-to-describe-images-with-natural-language-feedback`. 2.1

Sebastian Löbner. Quantification as a major module of natural language semantics. *Studies in discourse representation theory and the theory of generalized quantifiers*, 8:53, 1987. 4.1

Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research*, 11(Feb):955–984, 2010. 2.1, 4.1, 4.4, 4.4.1

Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. *arXiv preprint arXiv:1206.6423*, 2012. 6.1

Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. Learning to parse natural language commands to a robot control system. In *Experimental Robotics*, pages 403–415. Springer, 2013. 2.1

Scott Miller, David Stallard, Robert Bobrow, and Richard Schwartz. A fully statistical approach to natural language interfaces. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 55–61. Association for Computational

Linguistics, 1996. 2.2

Raymond J Mooney. Learning language from its perceptual context. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 5–5. Springer, 2008. 2.2

Linda M Moxey and Anthony J Sanford. Prior expectation and the interpretation of natural language quantifiers. *European Journal of Cognitive Psychology*, 5(1):73–91, 1993. 4.1

Karthik Narasimhan, Tejas D Kulkarni, and Regina Barzilay. Language understanding for textbased games using deep reinforcement learning. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Citeseer, 2015. 2.2

Stephen E Newstead and Janet M Collis. Context and the interpretation of quantifiers of frequency. *Ergonomics*, 30(10):1447–1462, 1987. 4.5

Sue Taylor Parker and Kathleen Rita Gibson. A developmental model for the evolution of language and intelligence in early hominids. *Behavioral and Brain Sciences*, 2(3): 367–381, 1979. 1

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*, 2015. 6.1

Jean Piaget, Marjorie Trans Gabain, and Ruth Trans Gabain. *The language and thought of the child*. Routledge/Taylor & Francis Group, 1959. 1

Karl Pichotta and Raymond J Mooney. Learning statistical scripts with LSTM recurrent neural networks. In *AAAI*, 2015. 6.1

Ashwin Ram. A theory of questions and question asking. *Journal of the Learning Sciences*, 1(3-4):273–318, 1991. 7

Dan Roth. Incidental supervision: Moving beyond supervised learning. In *AAAI*, pages 4885–4890, 2017. 2.1

Dan Roth and Kevin Small. Interactive feature space construction using semantic informa-

tion. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 66–74, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-29-9. URL `http://dl.acm.org/citation.cfm?id=1596374.1596388`. 3.1

Deb Roy and Ehud Reiter. Connecting language to the world. *Artificial Intelligence*, 167 (1-2):1–12, 2005. 2.2

Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, England, 1994. 7.2.3

S. J. Russell. *Analogical and Inductive Reasoning*. PhD thesis, Stanford University, Stanford, CA, USA, 1987. UMI Order No. GAX87-07736. 3.1

Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25(27):79–80, 1995. 1.5.1

Roger C Schank and Robert P Abelson. Scripts, plans, goals and understanding: an inquiry into human knowledge structures. *Erlbaum*, 1977. 6.1

John R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969. 2.2, 6.1

Shashank Srivastava and Eduard H. Hovy. Vector space semantics with frequency-driven motifs. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 634–643, 2014. 3

Shashank Srivastava, Dirk Hovy, and Eduard Hovy. A walk-based semantically enriched tree kernel over distributed word representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1411–1416, 2013. 3

Shashank Srivastava, Amos Azaria, and Tom Mitchell. Parsing natural language conver-

sations using contextual cues. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4089–4095. AAAI Press, 2017a. 6, 8.3.4

Shashank Srivastava, Igor Labutov, and Tom Mitchell. Joint concept learning and semantic parsing from natural language explanations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1527–1536, 2017b. 3, 4.1, 5, 8.3.4

Shashank Srivastava, Igor Labutov, and Tom Mitchell. Learning classifiers from declarative language. In *NIPS Workshop on Learning from Limited Data*, 2017c. 4, 8.3.4

Shashank Srivastava, Igor Labutov, and Tom Mitchell. Zero-shot learning of classifiers from natural language quantification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 306–316. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/P18-1029`. 4, 8.3.4

Mark Steedman and Jason Baldridge. Combinatory categorial grammar, 2011. 2.2

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Curran Associates, Inc., 2015. URL `http://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf`. 2.1

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth J. Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In Wolfram Burgard and Dan Roth, editors, *AAAI*. AAAI Press, 2011. 2.1

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. 1

Teun Adrianus Van Dijk. Text and context: Explorations in the semantics and pragmatics of discourse. *Nordic Journal of Linguistics*, 2, 1980. 2.2, 6.1

Lev Semenovich Vygotsky. *Mind in society: The development of higher psychological processes*. Harvard university press, 1980. 1

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 4.3.3

Sida Wang and Christopher D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 90–94, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=2390665.2390688`. 3.4.1

Sida I. Wang, Percy Liang, and Christopher D. Manning. Learning language games through interaction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016. URL `http://aclweb.org/anthology/P/P16/P16-1224.pdf`. 2.1

Sida I Wang, Samuel Ginn, Percy Liang, and Christopher D Manning. Naturalizing a programming language via interactive learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 929–938, 2017. 2.1

Yushi Wang, Jonathan Berant, and Percy Liang. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1332–1342, 2015. 6.1

Jason E Weston. Dialog-based language learning. In *Advances in Neural Information*

Processing Systems, pages 829–837, 2016. 2.1

Benjamin Lee Whorf. Science and linguistics. *Technology Review*, 1940. 1

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. The dialog state tracking challenge. In *SIGDIAL*, pages 404–413, 2013. 6.1

Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972. 2.1

Ludwig Wittgenstein. *Philosophical investigations. Philosophische Untersuchungen.* Macmillan, 1953. 1

Yuk Wah Wong and Raymond J Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. In *ACL*, volume 45, page 960, 2007. 6.1

Ilker Yildirim, Judith Degen, Michael K Tanenhaus, and T Florian Jaeger. Linguistic variability and adaptation in quantifier meanings. In *CogSci*, 2013. 4.1

John M Zelle and Raymond J Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055, 1996. 2.2, 3

Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pages 658–666, 2005. 2.2, 3, 3.2.1, 6.1

Luke S Zettlemoyer and Michael Collins. Online learning of relaxed ccg grammars for parsing to logical form. In *EMNLP-CoNLL*, pages 678–687, 2007. 2.2, 3.2.1, 4.2.1, 6.2.1, 6.3.1

Luke S. Zettlemoyer and Michael Collins. Learning context-dependent mappings from sentences to logical form. In *EMNLP-AFNLP*, ACL '09, pages 976–984, 2009. 6.1