Generative Models of Grounded Language Learning with Ambiguous Supervision

Joohyun Kim
Department of Computer Science
University of Texas at Austin
Austin, TX 78712
scimitar@cs.utexas.edu

Doctoral Dissertation Proposal

Supervising Professor: Raymond J. Mooney

Abstract

"Grounded" language learning is the process of learning the semantics of natural language with respect to relevant perceptual inputs. Toward this goal, computational systems are trained with data in the form of natural language sentences paired with relevant but ambiguous perceptual contexts. With such ambiguous supervision, it is required to resolve the ambiguity between a natural language (NL) sentence and a corresponding set of possible logical meaning representations (MR). My research focuses on devising effective models for simultaneously disambiguating such supervision and learning the underlying semantics of language to map NL sentences into proper logical forms. Specifically, I will present two probabilistic generative models for learning such correspondences. The models are applied to two publicly available datasets in two different domains, sportscasting and navigation, and compared with previous work on the same data.

I will first present a probabilistic generative model that learns the mappings from NL sentences into logical forms where the true meaning of each NL sentence is one of a handful of candidate logical MRs. It simultaneously disambiguates the meaning of each sentence in the training data and learns to probabilistically map a NL sentence to its MR form depicted in a single tree structure. Evaluations are performed on the RoboCup sportscasting corpous, which show that it outperforms previous methods.

Next, I present a PCFG induction model for grounded language learning that extends the model of Börschinger, Jones, and Johnson (2011) by utilizing a semantic lexicon. Börschinger et al.'s approach works well when there is limited ambiguity such as in the sportscasting task, but it does not scale well to highly ambiguous situations when there are large sets of potential meaning possibilities for each sentence, such as in the navigation instruction following task studied by Chen and Mooney (2011). Our model overcomes such limitations by employing a semantic lexicon as the basic building block for PCFG rule generation. Our model also allows for novel combination of MR outputs when parsing novel test sentences.

For future work, I propose to extend our PCFG induction model in several ways: improving the lexicon learning algorithm, discriminative re-ranking of top-k parses, and integrating the meaning representation language (MRL) grammar for extra structural information. The longer-term agenda includes applying our approach to summarized machine translation, using real perception data such as robot sensorimeter and images/videos, and joint learning with other natural language processing tasks.

Contents

1 Introduction

Understanding and learning the semantics of natural language is one of the long-standing, ultimate goals of artificial intelligence and natural language processing. "Language grounding", a process of mapping natural language to relevant aspects of the perceptual environment, is one approach towards this goal. Ideally, language grounding systems can mimic the language learning process of humans. A human child "grounds" language to perceptual contexts via repetitive exposure to co-occurrence of language and perception, which means humans also learn language in a *statistical* manner (Saffran, Johnson, Aslin, & Newport, 1999; Saffran, 2003). Attempts of modeling such grounded learning of humans have long been studied by a number of previous work (Bailey, Feldman, Narayanan, & Lakoff, 1997; Roy, 2002; Barnard, Duygulu, Forsyth, de Freitas, Blei, & Jordan, 2003; Yu & Ballard, 2004; Gold & Scassellati, 2007), where they tried to connect simplified natural language to raw perceptions. Other research has focused more on learning the meanings of full sentences while abstracting perceptions into machine-interpretable logical forms.

Conventional semantic parsing methods attempt to learn how to translate natural language to a formal, logical language (Zelle & Mooney, 1996; Zettlemoyer & Collins, 2005; Kate & Mooney, 2006; Wong & Mooney, 2007b; Zettlemoyer & Collins, 2007; Lu, Ng, Lee, & Zettlemoyer, 2008; Zettlemoyer & Collins, 2009). However, they require fully supervised corpora where one NL sentence is paired with one translated complete logical form. Constructing such parallel corpora can be difficult and time-consuming, even for human experts, and thus it is inherently difficult to extend such methods to large-scale systems or to apply them to other general problems. However, several recent approaches deal with more relaxed, natural settings of supervision where each NL sentence has a true meaning out of the several or exponentially many logical forms describing current perceptual states in the training data. My research focus follows this recent trend of grounded language learning methods. Particularly, I mainly investigate generative models that explain NL segments and logical components in a single hierarchical structure and learn how to map NL sentences into proper logical forms while simultaneously disambiguating such ambiguous supervision.

In this proposal, two different generative models for different levels of ambiguity are discussed as completed work. In Section 3, I will present a simultaneous alignment and semantic parsing model evaluated on the task of learning how to sportscast in virtual RoboCup 2D soccer games. The training data consist of natural language commentary to the ongoing game as well as automatically extracted logical forms representing the abstracted events currently happening. Thus, the training data has inherent ambiguity where each NL commentary has zero or one true meaning out of a several candidates of meanings. Chen and Mooney (2008) first attempted to solve this challenge via an EM-like retraining algorithm (Kate & Mooney, 2007). This approach is likely to suffer information loss, because in each retraining iteration, the model learns parameters from only the most probable MR match for each NL sentence. On the other hand, our model utilizes a generative model so that it probabilistically selects correct alignment as well as subsequent components of logical forms and natural language words and retains probabilistic counts for such relationships. Our approach is capable of disambiguating the match between language and meanings while also learning a complete semantic parser for mapping sentences to logical forms. Evaluation results on the Robocup domain show that our approach outperforms previous results on the NL–MR alignment task and language generation and also produces competitive performance on semantic parsing.

Next, I will present an unsupervised PCFG induction model evaluated on the previously investigated navigation task of Chen and Mooney (2011) in Section 4. The navigation task involves a much higher level of ambiguity. Each instruction is paired with a formal *landmarks plan* that includes a full description of the observed actions and world-states that result when someone follows this instruction. The main challenge here is that the instruction refers to only a subset of this full description, which inevitably results in

exponentially many potential alignments between each NL instruction and its logical form. Our model is a novel enhancement of an existing grounded language learning approach using unsupervised induction of *probabilistic context free grammars* (PCFGs) (Börschinger et al., 2011). Our model uses semantic lexemes as the basic building blocks for PCFG rules to avoid a combinatorial explosion in the number of matchings between components of NL words and logical forms. The semantic lexicon keeps the produced PCFG rule set to a tractable size compared to Börschinger et al.'s approach, while still exploiting full probabilistic predictions. Experimental results on the navigation corpus show the effectiveness of our approach in terms of partial parsing accuracy and end-to-end execution results.

For future work, I first propose and describe immediate short-term extensions of our lexicon-driven PCFG approach. Our approach relies on the quality of the learned semantic lexicon, since lexemes are the primitive components when learning the NL–MR mappings and for forming the final MR output. Thus, we propose to improve the current lexicon learning algorithm by integrating unsupervised part-of-speech (POS) learning inspired by the approach of Guo and Mooney(unpublished), which refines the lexicon by removing entries that violate the verb–action and object–argument criteria for NL–MR pairs. In addition, a discriminative re-ranking approach will be discussed next so that the final output is optimized. Top-k candidate parses will first be generated and then re-ranked discriminatively using additional features we can obtain from the parse tree structure. Moreover, I also plan to investigate how the structure of MRL grammar can help build better models instead of dealing with MRs as a combination of basic elements. Longer-term future agendas include application of our approach to summarized machine translation, extension to handle raw perception data such as extracted features from images/videos and real robot sensory data, aiding the learning model by joint learning with other NLP tasks such as information extraction and syntax parsing.

2 Background and Related Work

2.1 Grounded Language Learning

Grounded language learning is a process of learning the underlying meanings of natural language sentences in the context of observed perceptual environments. Ultimately, we want to build a computational system that can acquire language in a natural setting analogous to the way a human child learns language. A human child learns language through repetitive exposure to language along with the relevant perceptual object/event (Saffran et al., 1999; Saffran, 2003).

There have been many interesting research projects trying to build statistical learning systems that directly learns the meaning of language from raw visual perceptions (Bailey et al., 1997; Roy, 2002; Barnard et al., 2003; Yu & Ballard, 2004; Gold & Scassellati, 2007). However, such methods dealt with language of limited complexity and focused on the semantics of short phrases or words. Although it is ideal to learn the semantics of language in the form of raw perceptual data, those methods often lead to unnecessary complexity required for properly processing raw perceptions, thus limiting the effectiveness of the approaches. Therefore, some researchers have instead introduced "intermediate" logical representation which abstracts raw perceptual inputs into more compact, easily interpretable formats. Now then we have two separate problems instead of one grounded language learning problem: learning the underlying semantics of natural language in terms of well-formed logical representations and obtaining proper logical representations from raw perceptions such as images, videos, or sensor data. The latter part belongs to the area of computer vision or robotics, which itself is a challenging field of research. Our focus in this proposal is the former part, which falls in the field of semantic parsing and its reverse, language generation. While natural language is difficult for machines to understand because of its inherent structural and semantic ambiguity, logical

representations are self-explanatory, unambiguous, and well-formed so that they can be easily executed or used by computers. In the rest of this proposal, I will describe the completed work and future directions regarding how to find the relationships and translations between natural language and logical representations of perceptual contexts.

2.2 Learning for Semantic Parsing and Language Generation

Semantic parsing is the task of converting natural language sentences into appropriate logical forms that are easily interpreted by machines. While syntax parsing discovers inherent structure in natural language, semantic parsing aims to find non-trivial structural relationship between natural language words and elements of logical forms. The challenge mainly comes from the fact that natural language and logical forms do not always share structural similarity. Some words may correspond to a certain component of a logical form, while others may not have a true matching element. In this sense, semantic parsing is a challenging task and has long been studied by many researchers in the natural language processing community.

Conventional semantic parsing approaches learn to map natural language (NL) sentences to formal representations of their meaning (MR) via fully supervised training data consisting of NL/MR pairs (Zelle & Mooney, 1996; Zettlemoyer & Collins, 2005; Kate & Mooney, 2006; Wong & Mooney, 2007b; Zettlemoyer & Collins, 2007; Lu et al., 2008; Zettlemoyer & Collins, 2009; Kwiatkowski, Zettlemoyer, Goldwater, & Steedman, 2010). Such human annotated corpora are very expensive and difficult to build even for human experts, thus limiting the effectiveness of such conventional methods. Kate and Mooney (2007) extended one such conventional semantic parser learner, KRISP (Kate & Mooney, 2006), to work with more relaxed supervision. The extension, KRISPER, learns from ambiguous training data where one NL sentence has a true meaning out of a small set of multiple MRs. This relaxation reflects a more natural and general learning environment. Instead of annotating each sentence with a complete logical form one by one manually, it is possible to learn the meaning of natural language sentences from automatically extracted logical representations that reflect the perceptual context. The EM-like retraining algorithm of KRISPER alternates between finding the most probable one-to-one NL-MR matches based on parameters of the current iteration and updating the semantic parser with better estimates of the correct matches.

Language generation is the reverse process of semantic parsing, which is to learn how to translate from MR to NL. Many recent systems solve this problem in the context of chart generation (Kay, 1996). Carroll, Copestake, Flickinger, and Poznanski (1999) and Carroll and Oepen (2005) proposed a chart generator for Head-Driven Phrase Structure Grammar (HPSG), while White and Baldridge (2003), White (2004), and White (2006) used Combinatory Categorial Grammar (CCG) for natural language generation. However, these chart generation-based systems only focused on how to properly order the NL words to make sensible sentences, not on the relationship between NL words and MR elements so the meanings can be realized. Wong and Mooney (2007b) proposed a natural language generation system called WASP^{-1} based on the semantic parsing system WASP (Wong & Mooney, 2006). WASP is based on syntax-based statistical machine translation techniques. It induces a probabilistic synchronous context-free grammar (PSCFG) (Aho & Ullman, 1972) for generating corresponding NL–MR pairs. Since a PSCFG is symmetric with respect to the two languages it generates, the same learned model can be used for both semantic parsing (mapping NL to MR) (WASP) and natural language generation (mapping MR to NL) (WASP^{-1}). Since there are no pre-specified formal grammars for NL, WASP^{-1} learns an n-gram language model for the NL side and uses it to choose the most probable NL translation for a given MR using a noisy-channel model.

2.3 Content Selection

Before generating an NL sentence from an MR with a language generation model, we first need to decide which MR to describe. This process of selecting which MR to say is called *content selection* or *strategic generation*. It is the process of choosing *what to say*; as opposed to *surface realization* or *tactical generation*, which determines *how to say it*.

Chen and Mooney (2008) introduced the Iterative Generation Strategy Learning (IGSL) method for determining which event types a human commentator is more likely to describe in natural language. IGSL uses an EM-like method to train on ambiguously supervised data and iteratively improve probability estimates for each event type, specifying how likely each MR predicate is to elicit a comment. The algorithm alternates between two processes: calculating the expected probability of an NL–MR matching based on the currently learned estimates, and updating the probability of each event type based on the expected match counts. IGSL was shown to be quite effective at predicting which events in a RoboCup game a human would comment upon.

There are some other prior work regarding content selection. Zaragoza and Li (2005) tried to solve the problem using reinforcement learning in a video game environment where the speaker's goal is to help the listener reach the destination. An optimal strategy is found such that it conveys the most appropriate information. In addition, Barzilay and Lapata (2005) approached content selection as a collective task. Consistently better output is achieved by considering all the content selection decisions jointly and finding dependencies between each uttered items.

2.4 Learning from Ambiguous Supervision

Conventional supervised settings for semantic parsing or language generation are not suitable for general purpose domains or large-scale tasks. Manually annotating each NL sentence with a complete MR is often prohibitively expensive for certain tasks. Instead, it is more desirable to train models on natural supervision where the meaning of a sentence can be explained by some subpart of the surrounding perceptual context that is automatically extracted. This kind of ambiguous supervision normally appears in the form of training data where each NL sentence is paired with a number of candidate MRs.

Given such ambiguous supervision, the important challenge to solve is finding the true semantic alignment of NL–MR out of many possibilities in order to learn effective semantic parsers or language generators. Snyder and Barzilay (2007) proposed an alignment model between texts of American football game summaries with database entries containing statistics and events regarding the game and the football players. However, their approach uses direct supervision of the correct correspondence between the text and the database records. On the other hand, Liang, Jordan, and Klein (2009) proposed a probabilistic generative approach that produces a Viterbi alignment between NL and MRs. They use a hierarchical semi-Markov generative model that first determines which facts to discuss and then generates words from the predicates and arguments of the chosen facts. However, they only addressed the alignment problem and are unable to parse new sentences into meaning representations or generate natural language from logical forms.

Several recent works (Kate & Mooney, 2007; Chen & Mooney, 2008; Chen, Kim, & Mooney, 2010; Börschinger et al., 2011) tried to solve this semantic alignment problem in conjunction with semantic parsing. EM-like retraining algorithms such as KRISPER and WASPER first trains an initial semantic parser from the ambiguous training data by pairing each sentence with each of its candidate MRs. Then, the trained parser is used to select the most probable MR out of all the candidates. The algorithms iteratively improve the accuracies of both the semantic parser and the semantic alignment between NL–MR (Kate & Mooney, 2007; Chen & Mooney, 2008; Chen et al., 2010). On the other hand, Börschinger et al. (2011) proposed an

approach to convert such ambiguously supervised semantic parsing problem into a standard unsupervised PCFG induction problem. The PCFG rules are constructed so that each NL sentence is matched to a MR based on probabilistic rule weights learned through EM. Bordes, Usunier, and Weston (2010) viewed ambiguous supervision as a ranking problem. Since MR elements in the candidate sets are generally preferred, their approach learns a supervised ranking model that prefers those candidate MRs over other random MR elements.

In addition, a recent piece of research has investigated a higher degree of ambiguous supervision in the task of learning to follow navigation instructions in a virtual environment (Chen & Mooney, 2011). In this task, each NL instruction is paired with a full description of the observed actions and world-states that will be encountered while following the instruction. The challenging part of this task is that only a subpart of the entire observed perceptual context is relevant to a given NL instruction, thus leading to the problem of searching a combinatorial number of possible alignments between a NL sentence and all subsets of the descriptive MR. Instead of directly solving this combinatorially ambiguous supervision, Chen and Mooney circumvent this complexity by refining the whole contexts with a learned semantic lexicon. Then, the problem is reduced to standard supervised semantic parsing, but there are possible information losses while greedily selecting lexicon entries during the refinement process.

Other researchers recently attempted to learn semantic parsers given only a weak supervision of *responses* (Clarke, Goldwasser, Chang, & Roth, 2010; Liang, Jordan, & Klein, 2011). Formal meaning representations (MRs) are easily understood and used for execution by machines. Thus, we could utilize the responses instead of full MRs as a weak indication whether a certain intermediate MR output is correct or not during the learning process. This feedback drives the semantic parser learner toward more accurate internal parameter estimation. These response-driven semantic parsing models treat the formal MRs as latent variables to be estimated, and optimize the MR output for a novel NL sentence with respect to the known MR grammar structure, incorporating a small domain-specific knowledge. However, the response-based approaches may not be applied to the certain domains where MRs are the descriptions of actions or surrounding environments and cannot be evaluated to obtain the direct responses.

2.5 Other Related Works

One of the earliest work on grounded language learning is by Siskind (1996). His approach solves referential uncertainties of words by capturing how the same word is uttered with the same perception. However, this approach only solves the ambiguity for the semantics of words and does not provide any solution for compositional meanings.

Following Siskind, many researchers in robotics and computer vision tackled the problem of learning the grounded meanings of natural language words or short phrases from raw perceptual contexts (Roy, 2002; Bailey et al., 1997; Barnard et al., 2003; Yu & Ballard, 2004). However, most of these works have limited complexity on the side of natural language since their major challenge came from how to describe and abstract raw image descriptions. Consequently, they do not consider or exploit syntactic or semantic structure of natural language while connecting it with perceptions. To the contrary, our completed work actively utilizes the common structure that entails both natural language and abstracted perceptions to understand the underlying semantics of language.

In addition, a recent system called TWIG (Gold & Scassellati, 2007) has been proposed and showed how word-learning process can be aided by existing knowledge about natural language. In their system, the robot learns the meaning of new referential words such as “I” and “you” by watching a catching game and inferring based on previous knowledge about related natural language phrases.

There have been a number of works on connecting natural language captions to images or videos in

computer vision (Barnard et al., 2003; Li & Wang, 2008; Li, Socher, & Fei-Fei, 2009; Wang, Blei, & Li, 2009; Gupta, Kim, Grauman, & Mooney, 2008; Fleischman & Roy, 2007; Gupta & Mooney, 2009). Their work are mainly concerned with how words or short phrases of image/video descriptions can help the task of image classification or automatic image annotation. Although their descriptions and extracted features are extensive about visual contents, their natural language is relatively limited and they do not utilize any linguistic structure, only using bag-of-words model.

There are also other recent approaches that solve grounded language learning in terms of interpreting natural language instructions in the specified environments. Some of them built systems that learn to map natural language instructions into executable commands for a robot navigating a real environment (Shimizu & Haas, 2009; Matuszek, Fox, & Koscher, 2010; Vogel & Jurafsky, 2010; Kollar, Tellex, Roy, & Roy, 2010; Tellex, Kollar, Dickerson, Walter, Banerjee, Teller, & Roy, 2011). Even though those approaches deal with full natural language sentences as linguistic inputs, their systems make the semantic learning task too simplified by ignoring relevant objects in the environment, or assuming predefined spatial words, direct matchings between NL words and the names of objects and other landmarks in the MR, and/or an existing syntactic parser. Other projects (Branavan, Chen, Zettlemoyer, & Barzilay, 2009; Branavan, Zettlemoyer, & Barzilay, 2010) learn to map natural language instructions to executable actions using reinforcement learning in Windows GUI and card game environments. However, they exploit language-specific cues that co-occur both in natural language instructions and words in the evaluated environments. In contrast, our systems do not assume any prior knowledge about the language itself and have to learn all the connections between natural language and logical representation, and thus do not depend on specific languages. In addition, their methods do not learn semantic parsers that map natural language instructions into full, semantic meaning representations.

3 Generative Alignment and Semantic Parsing for Learning from Ambiguous Supervision

In this section, we present a probabilistic generative model for learning semantic parsers that is trained on the ambiguous setting where NL sentences paired with world states consisting of multiple candidate logical MRs (Kim & Mooney, 2010). It disambiguates the meaning of each sentence while simultaneously learning a semantic parser that maps sentences into logical forms. Chen and Mooney (2008) introduced the problem of learning to sportscast by simply observing natural language commentary on simulated RoboCup robot soccer games. The 1-to-*N* NL–MR ambiguity of the training data poses a serious challenge to learning accurate semantic parsers or language generators. We need to learn the correct semantic alignment between NL and MR since the correct alignment of the training data is unknown.

The approach by Chen and Mooney (2008) retrains existing supervised semantic parsers iteratively on the disambiguated NL–MR pairs produced by the previous iteration. However, it suffers possible information loss since it does not run on a well-defined probabilistic model. On the other hand, Liang et al. (2009) proposed a probabilistic generative alignment model for ambiguous supervision. Despite its improved performance, the model is only capable of semantic alignment between NL–MR and does *not* learn either a semantic parser or a language generator. In addition, they assume a bag-of-words model for natural languages and do not incorporate linguistic syntax which includes additional cues to be exploited.

Our generative model overcomes some of the limitations of these previous methods and provides simultaneous semantic alignment and semantic parsing for ambiguous supervision using the Hybrid tree model proposed by Lu et al. (2008), which generates NL and MR components in a single tree structure. Experimental results on the sportscasting data show that our approach outperforms all the previous results on the

	English	Korean
# of NL comments	2036	1999
# of words	11742	7941
Average words per NL comment	5.77	3.97
# of extracted MR events	10452	10668
# of NLs with matching MRs	1868	1913
# of MRs with matching NLs	4670	4610
Average number of MRs per NL	2.50	2.41

Table 1: Statistics for Robocup sportscasting data

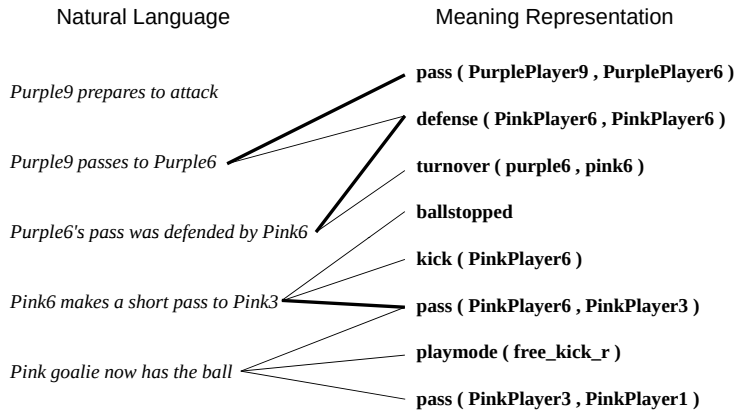


Figure 1: Sample trace from Robocup English data.

NL–MR matching (alignment) and language generation task and also achieves competitive performance on semantic parsing.

The RoboCup sportscasting data (Chen & Mooney, 2008; Chen et al., 2010) was collected by asking humans to commentate the 4 final games (2001 to 2004) of the Robocup simulation soccer league. Table 1 shows statistics about the data. Each NL commentary sentence is paired with automatically extracted MRs of ongoing simulation events that occurred in the previous 5 seconds (an average of 2.5 events).

Figure 1 shows a sample trace from the English data. As shown, each NL commentary sentence normally has several candidate MR matches that occurred within the 5-second window, indicated by edges between NL and MR. Bold edges denote gold standard alignment manually constructed solely for evaluation purposes. However, it is not guaranteed that every NL has a gold matching MR, because sometimes there are unrecognized or undetected events and sometimes there are NL commentaries that describe high level concepts about the game (e.g. the pink team is sloppy today). Such ambiguity brings the additional challenge that a given NL sentence may not have a correct matching MR.

3.1 Generative Hybrid Tree Model for Semantic Parsing

Our model is built on top of the generative semantic parsing model using a hybrid-tree framework (Lu et al., 2008). Moreover, our model has the additional capability of selecting which MR out of all the candidates will be described. A *hybrid tree* is defined over a pair, $(\mathbf{w}, \mathbf{m})$, of a NL sentence and its corresponding MR. The tree describes a correspondence of NL word segments and MR components following the grammatical

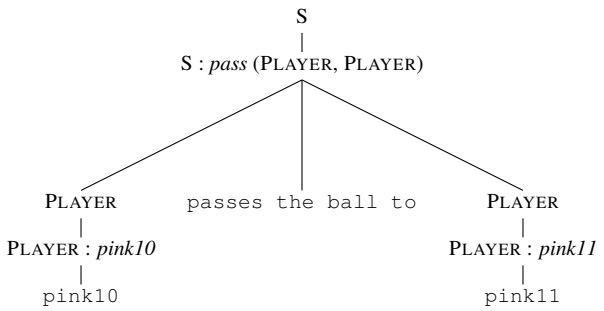


Figure 2: Sample hybrid tree of NL/MR pair from the English sportscasting dataset: PINK10 PASSES THE BALL TO PINK11 / *pass*(*pink10*, *pink11*)

structure of the MR. In a hybrid tree, MR production rules constitute the internal nodes, while NL words (or phrases) constitute the leaves. A sample hybrid tree from the English RoboCup data is shown in Figure 2.

A generative process based on hybrid trees is defined as follows: starting from a root semantic category, the model generates a production of the MR grammar, and then subsequently generates a mixed hybrid pattern of NL words and child semantic categories. This process continues until all the leaves in the hybrid tree become NL words. The generation assumes a Markov process, implying that each step is only dependent on its parent step.

Lu et al.'s generative parsing model estimates the joint probability $P(\mathscr{T}, \mathbf{w}, \mathbf{m})$, the probability of generating a hybrid tree $\mathscr{T}$ with NL $\mathbf{w}$, and MR $\mathbf{m}$. This probability is calculated by the whole product of the probabilities of all the generation steps in the tree. The data likelihood of the pair $(\mathbf{w}, \mathbf{m})$ given by the learned model becomes the sum of $P(\mathscr{T}, \mathbf{w}, \mathbf{m})$ over all the possible hybrid trees, because a hybrid tree for a NL $\mathbf{w}$ and a MR $\mathbf{m}$ is not unique.

The model runs in conventional, fully supervised settings. In order to learn from ambiguous supervision, we extend this model to include an additional generative process for selecting the subset of available MRs used to generate NL sentences.

3.2 Our Generative Model for Alignment and Language Grounding

Our model estimates $P(\mathbf{w}|\mathbf{s})$, where $\mathbf{w}$ is a NL sentence and $\mathbf{s}$ is a world state consisting of several candidate MRs matched to $\mathbf{w}$. In this setting, our approach is intended to support both determining the most likely match between a NL and its MR, **and** semantic parsing, i.e. finding the most probable mapping from a given NL sentence to a MR logical form.

Our generative model consists of two stages:

- Event selection: $P(\mathbf{e}|\mathbf{s})$, chooses the event $\mathbf{e}$ in the world state $\mathbf{s}$ to be described.
- Natural language generation: $P(\mathbf{w}|\mathbf{e})$, models the probability of generating natural-language sentence $\mathbf{w}$ from the MR specified by event $\mathbf{e}$.

3.2.1 Event selection

The event selection model specifies the probability distribution for picking an event that is likely to be commented upon amongst the multiple candidate MRs appearing in the world state $\mathbf{s}$. The probability of selecting an event is assumed to depend only on its event type as given by the predicate of its MR. For

example, the MR $pass(pink10, pink11)$ has the event type *pass* and arguments *pink*10 and *pink*11. The probability of picking an event e of type t_e is $p(t_e)$. If there are multiple type t events in $\mathbf{s}$, then a type t event is selected uniformly from the set $\mathbf{s}(t)$ of events of type t in $\mathbf{s}$. Thus, the probability of picking an event is given by:

$$P(\mathbf{e}|\mathbf{s}) = p(t_e)\frac{1}{|\mathbf{s}(t_e)|} \tag{1}$$

This model is similar to the *record choice* model of Liang et al. (2009), but it is only modeling *salience* such that some event types are more likely than others. Our model does not consider the order of event types (*coherence*) because the RoboCup sportscasting data only has at most one true MR for a given NL sentence.

3.2.2 Natural language generation

The natural-language generation model defines the probability distribution of NL sentences given a MR specified by the previously selected event in the event selection model. We use Lu et al. (2008)'s generative semantic parsing model for this step:

$$P(\mathbf{w}|\mathbf{e}) = \sum_{\forall \mathscr{T} \ over \ (\mathbf{w},\mathbf{m})} P(\mathscr{T}, \mathbf{w}|\mathbf{m}) \tag{2}$$

where $\mathbf{m}$ is the MR defined by event $\mathbf{e}$ and $\mathscr{T}$ is a hybrid tree defined over the NL–MR pair $(\mathbf{w}, \mathbf{m})$.

The probability $P(\mathscr{T}, \mathbf{w}|\mathbf{m})$ is given by the generative semantic parsing model (Lu et al., 2008) with the inside probability of the NL–MR pair $(\mathbf{w}, \mathbf{m})$. The likelihood of a sentence $\mathbf{w}$ is then the sum over all possible hybrid trees defined by the NL–MR pair $(\mathbf{w}, \mathbf{m})$. We used their bigram model out of the three proposed models (i.e. unigram, bigram, and mixgram) which are categorized by whether a NL word or a semantic category is dependent upon the previously generated one. In our experiments, the bigram model always performed the best on all tasks and thus we use the bigram model.

The natural language generation model replaces the *field choice* model and *word choice* model of Liang et al. (2009) in the semantic alignment task. It also considers the order of predicates and arguments in a MR as well as the generation of NL words and phrases, since the model constructs an ordered hybrid tree structure that generates NL words, MR semantic categories, and MR grammar rules.

3.3 Learning

Conventional EM methods are used to train our generative model. The process is similar to Lu et al.'s model, an inside-outside style algorithm that generates a hybrid tree from the NL–MR pair $(\mathbf{w}, \mathbf{m})$, but our model additionally considers expected counts under the posterior $P(\mathbf{e}|\mathbf{w}, \mathbf{s}; \theta)$ in the E-step and normalizes the counts in the M-step. Training time takes about 30 minutes to run for sportscasts of three games in either the English or Korean dataset.

However, the experiments show that EM method tends to fall into local optima when estimating the event-type selection probabilities, $p(t)$, thus degrading the overall performance. To resolve this issue, we initialized the parameters with the corresponding strategic generation values learned by the IGSL method of Chen and Mooney (2008). IGSL priors serve as a good starting point for our event selection model since IGSL has already been shown to effectively predict which event types are likely to be described compared to actual human comments in sportscasting data (Chen & Mooney, 2008).

The generative semantic parsing model by Lu et al. (2008) is trained through several stages to provide the best performing results. The bigram model we used in our model was trained on the basis of parameters

Systems	Matching	Semantic Parsing	NL Generation	Ambig. Training
Our model	√	√	-	√
Liang et al. (2009)	√	-	-	√
Lu et al. (2008)	-	√	-	-
WASP^{-1} (Wong & Mooney, 2007b)	-	√	√	-

Table 2: Overview of various systems and models used in the experiments. Each column indicates the capability on various tasks.

previously learned for the IBM Model 1 (Brown, Della Pietra, Della Pietra, & Mercer, 1993) and unigram model. We followed a similar learning strategy for performance. The multiple learning stages led to vulnerability of the model getting stuck in local optima when running EM across these multiple steps. We also tried using random restarts with several initializations, but IGSL priors provided the best results in the evaluations.

3.4 Experimental Evaluation

For evaluation, we followed the same evaluation schemes as in Chen and Mooney (2008) covering 3 tasks: NL–MR matching (semantic alignment), semantic parsing, and surface realization. RoboCup sportscasting data contains 4 separate games and we performed leave-one-game-out (4-fold) cross validations using 3 games for training and the remaining 1 game for testing to evaluate semantic parsing and surface realization. Since the matching (semantic alignment) task is essentially disambiguating the training data, the performance of matching is evaluated on the training data.

The accuracy of matching and semantic parsing is measured using F-measure, the harmonic mean of precision and recall. Natural language generation is evaluated using BLEU score (Papineni, Roukos, Ward, & Zhu, 2002) between the generated sentences and the reference NL sentences in the test set. Systems we compared to include Chen and Mooney (2008) and Chen et al. (2010), and for alignment results, Liang et al. (2009) is also included.

In Table 2, we present various systems and models used in the experimental evaluations and their capabilities for the tasks. Our model is capable of learning semantic parsers while disambiguating the correct matching of training data.

3.4.1 NL–MR Matching (Semantic Alignment)

The matching or semantic alignment task measures how well the system finds the correct NL–MR alignment out of ambiguous examples consisting of a NL sentence and multiple potential MRs. As described earlier, training examples in the RoboCup sportscasting data have up to one correct matching. Our model outputs the most probable matching as a NL **w** and a MR **m** if and only if **m** is the most probable parse of **w** according to the learned semantic parser. Thus, our model does not force every NL to match to a MR. Some NL sentences whose most probable parse is not one of the candidate MRs are left unmatched. Matching output is evaluated against the manually constructed gold-standard matches, which is never used during training.

Evaluation results for the English and Korean datasets are shown in Table 3. Since the Korean data was not yet available for use by either Chen and Mooney (2008) or Liang et al. (2009), we cited those results from Chen et al. (2010). Our best approach outperforms all previous methods by large margins when

	English	Korean
Chen and Mooney (2008)	0.681	0.753
Liang et al. (2009)	0.757	0.694
Chen et al. (2010)	0.793	0.841
Our model	0.832	0.800
Our model with IGSL prior initializations	**0.885**	**0.895**

Table 3: NL–MR Matching Results (F-measure).

	English	Korean
Chen and Mooney (2008)	0.702	0.720
Chen et al. (2010)	0.803	**0.812**
Our learned parser	0.742	0.764
Lu et al. (2008) initialized with our model's matching	**0.810**	0.794
Lu et al. (2008) initialized with Liang et al. (2009)	0.790	0.690
WASP initialized with our model's matching	0.786	0.808
WASP initialized with Liang et al. (2009)	0.803	0.740

Table 4: Semantic Parsing Results (F-measure).

using IGSL priors. In particular, our model also outperforms the generative alignment model by Liang et al. (2009), implying that the extra linguistic information and MR grammatical structure result in a more effective model than a Markov model with a bag-of-words model.

3.4.2 Semantic Parsing

Semantic parsing is evaluated by how accurately the systems predict mapping novel NL sentences into their proper corresponding MRs in the test data. Table 4 presents the results. We compare to the best results presented in the cited papers: WASPER-GEN for Chen and Mooney (2008), WASPER with Liang et al.'s matching initialization for English and WASPER-GEN-IGSL-METEOR with Liang et al.'s initialization for Korean as the results for Chen et al. (2010). Semantic parsing results with our directly learned parser from the ambiguous training data are presented, as well as supervised parsers (both WASP and Lu, Ng, and Lee's) trained on the NL–MR matching output by our model.[1] For additional comparisons, Lu et al.'s parser and WASP trained on Liang et al.'s NL–MR matchings are also presented.

Our initial learned semantic parser performs better than Chen and Mooney (2008), but worse than Chen et al. (2010). Training WASP and Lu et al.'s parsers on our highly accurate NL–MR matchings improved the results over Liang et al.'s matchings. It is also noteworthy that retraining on the hardened one-to-one supervision of the most probable NL–MR matches gives better performance than the parser directly trained using EM. The uncertainty caused by incorrect NL–MR matchings seems to affect the overall parsing performance.

[1] All our semantic parsing results used IGSL initialization, which results in the best performances.

	English	Korean
Chen and Mooney (2008)	0.4560	0.5575
Chen et al. (2010)	0.4599	0.6796
WASP^{-1} trained on matching of Liang et al. (2009)	0.4580	0.5828
WASP^{-1} trained on our matching outputs	**0.4727**	**0.7148**

Table 5: Surface realization results (BLEU score).

3.4.3 Surface Realization

The surface realization or tactical generation task evaluates how well a system generates accurate NL sentences from novel test MRs. Since our semantic parsing model does not support surface realization, the reverse task, we trained the publicly available WASP^{-1} system (Wong & Mooney, 2007a) on our disambiguated NL–MR matches. Since we are using WASP^{-1}, we can directly compare with the results of Chen and Mooney (2008) and Chen et al. (2010).

Table 5 shows the surface realization results of our model and the best reported results from the cited papers: WASPER-GEN for Chen and Mooney (2008), WASPER trained with Liang et al.'s matching for the English results of Chen et al. (2010), and WASPER-GEN with Liang et al.'s initialization for the Korean dataset. WASP^{-1} trained on our NL–MR matching results performed the best. It should also be noted that WASP^{-1} trained with our matchings performs better than WASP^{-1} trained with Liang et al.'s matchings.

3.5 Discussions

Overall, our model performs particularly well at the matching task. However, improved matching does not transfer to notably better semantic parsing results, considering there was a 10 percentage improvement for matching compared to a 1 point improvement on the semantic parsing task.

This seems to be due to the nature of the noise in the matching results. Although Liang et al.'s alignment model gives a much lower F-measure, its matches have higher precision and contain fewer noisy, misleading NL–MR pairs, while our model's precision is relatively low compared to the F-measure. Our model predicts some misleading matches when the gold standard match does not exist, resulting in worse semantic parsers due to the noisy NL–MR training pairs.

Compared to Liang et al. (2009), our more accurate matchings provide a clear improvement in both semantic parsing and surface realization. However, the difference in semantic parsing seems to be less than that in surface realization. As discussed by Chen and Mooney (2008) and Chen et al. (2010), this difference seems to be because surface realization is somewhat easier than semantic parsing in that semantic parsing needs to learn to map a variety of synonymous NL sentences to the same MR, while surface realization only needs to learn one way to produce a correct NL description of a MR.

4 Unsupervised PCFG Induction for Grounded Language Learning with Highly Ambiguous Supervision

In this section, I will discuss our unsupervised *probabilistic context-free grammar* (PCFG) induction model for learning the semantics of language when the training data is highly ambiguous (Kim & Mooney, 2012). Particularly, we focus on the navigation task (Chen & Mooney, 2011) where the goal is to interpret natural

language instructions in virtual environments so that an agent can perform the desired actions. The navigation task deals with a much higher level of ambiguity compared to the RoboCup sportscasting task discussed in Section 3. The RoboCup sportscasting task only requires the system to find the correct matching for a NL commentary sentence out of several potential MR events observed within the past 5 seconds. On the contrary, the navigation task requires the system to disambiguate the training data where each instructional sentence is paired with a formal *landmarks plan* (represented in a large graph structure) that includes a full description of the observed actions and world-states that are encountered while following the instruction. The major challenge comes from the fact that the NL instruction refers to only a subgraph of the formal landmarks plan. This inevitably leads to a combinatorial number of possible meanings when finding a true match for a given sentence.

To resolve this problem, I present a novel enhancement of the unsupervised *probabilistic context free grammar* (PCFG) induction method for grounded language learning introduced by Börschinger et al. (2011). Börschinger et al.'s approach works for relatively simple ambiguous supervision where there is up to one true meaning out of a small set of contextual meanings for a NL sentence, which is the case for the RoboCup sportscasting task. Their approach first constructs a large set of production rules from the ambiguous training set of a NL sentence paired with multiple MRs, and then optimizes the weights of the PCFG grammar using EM. Parsing a novel sentence with this learned grammar produces a parse tree containing the formal MR parse in the top nonterminal. Although this approach is effective for simple ambiguous supervision such as the sportscasting data, applying it to problems with highly ambiguous supervision such as the navigation task leads to a prohibitively large number of PCFG production rules. For instance, there are a number of training examples in the navigation data containing more than 20 actions for a single NL instruction sentence, which produces more than 20! ($> 10^{18}$) PCFG production rules to train on.

To overcome such difficulty, our approach enhances Börschinger et al.'s model by using semantic lexemes as the basic building block when constructing PCFG production rules. While Börschinger et al. used each MR constituent to probabilistically generate NL words, our approach uses lexemes (pairs of a MR graph and a NL phrase) which form meaningful semantic concepts. The advantage of this enhancement is that we directly connect semantic concepts to corresponding NL words during training. Also, the semantic concepts represented by lexeme MRs will intuitively form hierarchical structure analogous to syntactic hierarchy in syntax parsing. In addition, our approach is able to solve some of the limitations of Börschinger et al.'s model in that the number of PCFG production rules remains tractable since semantic lexemes as basic units encode MRs in compact representations for complicated MR languages. Moreover, our model can also produce novel final MR parses which were never seen during training, whereas Börschinger et al. is not.

4.1 Unsupervised PCFG Induction for Grounded Language Learning

Börschinger et al. (2011) introduced an unsupervised PCFG induction model for grounded language learning. It automatically constructs a PCFG that generates natural language (NL) sentences from formal meaning representations (MRs). The nonterminals in the grammar correspond to complete MRs and MR constituents, while NL phrases and words are expressed as terminals. The generative process of PCFG describes how a composite MR generates its MR constituents, followed by each constituent generating NL words eventually. First, the nonterminal for a composite MR generates each of its MR constituents. Since we do not know the order in which each constituent will generate NL words, every possible permutation of the constituents must be included in order to consider all the possibilities. Second, the nonterminal for a MR constituent generates $Phrase_x$, representing a sequence of NL words connected to the constituent x. $Phrase_x$ is then used to generate a sequence of $Word_x$ which subsequently produces NL words. By training the Inside-Outside

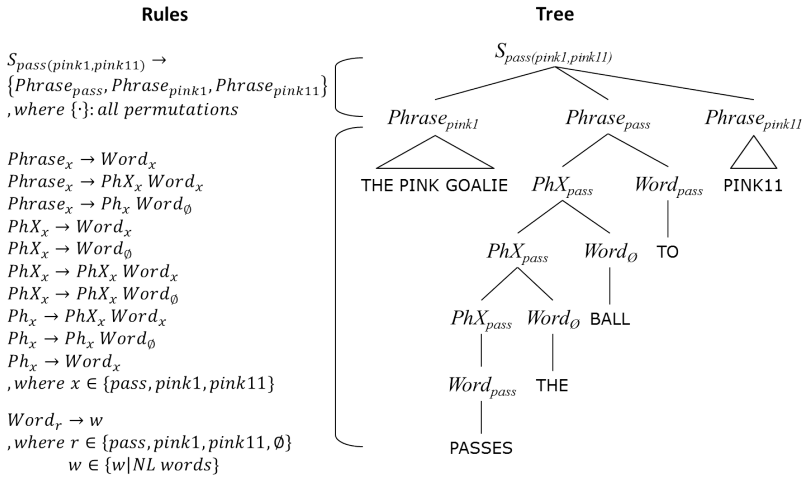


Figure 3: Derivation tree for the NL/MR pair: THE PINK GOALIE PASSES THE BALL TO PINK11 / $pass(pink1, pink11)$. Left side shows PCFG rules that are added for each stage (full MR to MR elements, and MR elements to NL words).

algorithm on the produced PCFG rules, the system learns the probabilistic relationships between NL words, MR constituents, and complete MRs by weighing the rules. Figure 3 shows a derivation tree for a sample NL-MR pair and the PCFG rules that are constructed for it. When parsing a novel sentence into the most probable parse tree, we are able to get the most likely MR interpretation for the sentence by reading the top nonterminal containing the MR.

However, this approach has several clear limitations. First, it only works for finite MR languages, and the produced PCFG becomes intractably large even for finite but moderately complex MRs. In addition, it is only able to produce MRs previously seen during training as the parse result of a novel NL sentence. On the contrary, our approach uses a semantic lexicon to constrain the space of productions, thereby keeping the PCFG tractable even for complex MRs. Also, it has the ability to compose novel MRs when parsing test sentences as well as handle infinite MR languages. Finally, our approach works with a much higher level of ambiguous supervision where a NL sentence refers to only some subset of a matching MR representation, which implies an exponential number of possibilities for the true matching.

4.2 Navigation Task and Dataset

For evaluating our model, we employ the task and data introduced by Chen and Mooney (2011) where the goal is to interpret and follow NL navigation instructions in a virtual world by simply observing how humans follow them. Figure 4 shows a sample execution path in a particular virtual world. Given the training data of the form $\{(e_1, a_1, w_1), \ldots, (e_n, a_n, w_n)\}$, where e_i is a NL instruction, a_i is an observed action sequence, and w_i is the current world state (patterns of floors and walls, positions of any objects, etc.), we want to produce the correct actions a_j for a novel (e_j, w_j).

In order to learn, the task requires us to infer the intended formal plan p_i (the MR for a sentence in this task) which produced the action sequence a_i from the instruction e_i. However, there is a large number of possibilities when choosing a formal plan for any given action sequence. For a simple example, there are several ways to describe the actions of going two steps toward a sofa and then turning right. In a straightforward manner, we can describe the actions as `Forward(steps:2), Turn(RIGHT)` by just describing the

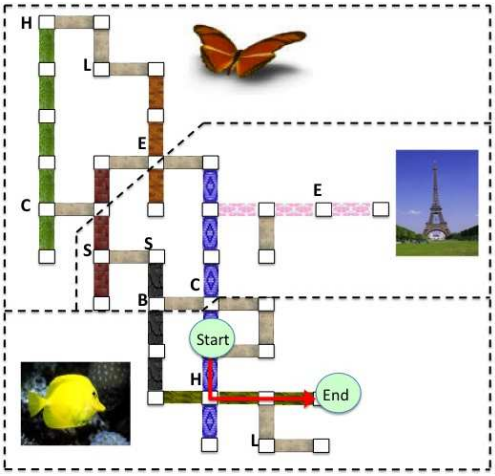


Figure 4: Sample virtual world from Chen and Mooney (2011) of interconnecting hallways with different floor and wall patterns and objects indicated by letters (e.g. "H" for hatrack).

atomic actions, or $\texttt{Forward}(), \texttt{Verify}(\texttt{front} : \texttt{SOFA}), \texttt{Turn}(\texttt{RIGHT})$ using notable objects along the way. Chen and Mooney (2011) called the former a *basic plan* and the latter a *landmarks plan*. They focused more on the landmarks plan since it carries more information for understanding the semantics of instructions and is closer to how humans actually describe navigational directions in the real world. Also, they showed that landmarks plans led to better overall evaluation performance with their proposed system.

Regarding their system, it first constructs a formal landmarks plan, c_i, for each a_i, which is a graph representing the context consisting of a full description of every action in the sequence and the world-state that is encountered while following the actions. The hard part is that the correct plan MR, p_i, is assumed to be a subgraph of c_i, which implies there is an exponential number of possibilities to choose a correct MR from. The landmarks and correct plans for a sample instruction are shown in Figure 5.

To circumvent this combinatorial problem, Chen and Mooney (2011) first learn a semantic lexicon that maps NL words and short phrases to small MRs (subgraphs) by finding correlations between NL phrases and MR subgraphs. The learning process is similar to other "cross-situational" approaches of learning word meanings (Siskind, 1996; Thompson & Mooney, 2003). From the training data (e_i, c_i), the algorithm first collects all navigation plans c_js co-occurring with an n-gram w as candidate meanings for w. This initial candidate meaning set is expanded while repeatedly taking intersections between the candidate meanings, where the intersections can be obtained by taking the largest common subgraphs. The resulting candidate set is ranked by the following scoring metric for an n-gram w and an MR graph m:

$$Score(w,m) = p(m|w) - p(m|\neg w)$$

, which measures how much more likely a MR m appears when w is present compared to when it is not.

After obtaining a lexicon, the *plan refinement* step estimates p_i from c_i by greedily selecting high-scoring *lexemes* (i.e. lexicon entries of (w_j, m_j)) whose phrases (w_j) cover the instruction e_i and introduce components (m_j) from the landmarks plan c_i. The refined plans are then used to train a semantic parser learner as a supervised training set (e_i, p_i). The trained semantic parser can parse a novel instruction into a formal plan, which is finally executed for end-to-end evaluation. Figure 6 illustrates the overall system.

As this figure indicates, our new PCFG method replaces the roles of the plan refinement step and the

Instruction: "at the easel, go left and then take a right onto the blue path at the corner"

Landmarks plan:
Travel (steps: 1) ,
Verify (**at: EASEL** , side: CONCRETE HALLWAY) ,
Turn (LEFT) ,
Verify (front: CONCRETE HALLWAY) ,
Travel (steps: 1) ,
Verify (side: BLUE HALLWAY , **front: WALL**) ,
Turn (RIGHT) ,
Verify (back: WALL , **front: BLUE HALLWAY** , front: CHAIR ,
front: HATRACK , left: WALL , right: EASEL)

Figure 5: Sample instruction with its landmarks plan. Bold components are the true plan.

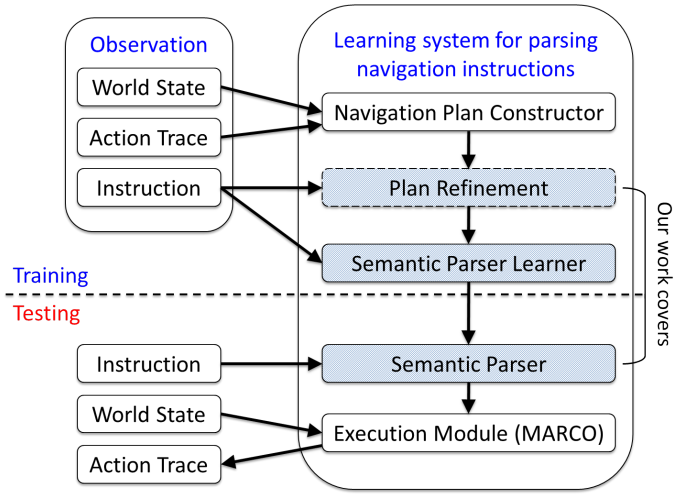


Figure 6: An overview of Chen and Mooney (2011)'s system. Our approach replaces the roles of the plan refinement component and the semantic parser.

semantic parser in their system. It is a unified system that simultaneously disambiguates the training data and learns a semantic parser. We use the landmarks plans and the learned lexicon produced by Chen and Mooney (2011) as inputs to our system.

4.3 Our PCFG Approach

Like Börschinger et al. (2011), our approach learns a semantic parser directly from ambiguous supervision, specifically NL instructions paired with the complete landmarks plans as context in the navigation dataset. Our method utilizes the semantic lexemes as basic building blocks to find correspondences between NL words and semantic concepts represented by the MRs in the lexemes, instead of building connections between NL words and each MR constituent as in Börschinger et al. (2011). Particularly, we utilize the hierarchical subgraph relationships between the MRs in the semantic lexicon to produce a smaller, more focused set of PCFG rules.[2] The intuition behind this is analogous to the hierarchical relations between syntactic

[2]Total number of PCFG rules for the training set is about 18,000. Note that Börschinger et al.'s method produces 33,000 rules for the simpler sportscasting domain.

Algorithm 1 LEXEME HIERARCHY GRAPH (LHG)

Input: Training instance (e_i, c_i), Lexicon L
Output: Lexeme hierarchy graph for (e_i, c_i)

Find relevant lexemes $(w_1^i, m_1^i), \ldots, (w_n^i, m_n^i)$ s.t. $m_j^i \subset c_i$
Create a starting node T; $MR(T) \leftarrow c_i$
for all m_j^i in the descending order of size **do**
    Create a node T_j^i; $MR(T_j^i) \leftarrow m_j^i$
    PLACELEXEME(T_j^i,T)
end for

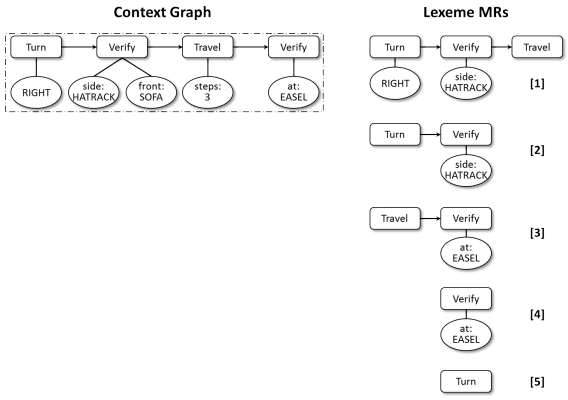
procedure PLACELEXEME(T',T)
    for all children T_j of T **do**
        if $MR(T') \subset MR(T_j)$ **then**
            PLACELEXEME(T',T_j)
        end if
    end for
    if T' was not placed under any child T_j **then**
        Add T' as child of T
    end if
end procedure

categories in syntax parsing. In syntax parsing, high level categories such as S, VP, or NP refer to bigger concepts which are further divided into smaller concepts such as V, N, or Det, therefore forming a hierarchical structure. Inspired by this notion, we introduce a directed acyclic graph called the *Lexeme Hierarchy Graph* (LHG) which represents the hierarchical relationships between lexemes. Since complex lexeme MRs represent complicated combined semantic concepts and simple MRs represent simple concepts, it is natural to construct a hierarchy between the lexeme MRs. The LHGs for all the training examples are used to construct production rules for the PCFG, which are then parametrized using EM. Finally, novel sentences are semantically parsed by computing their most-probable parses using the trained PCFG and extracting an MR from the resulting parse tree.

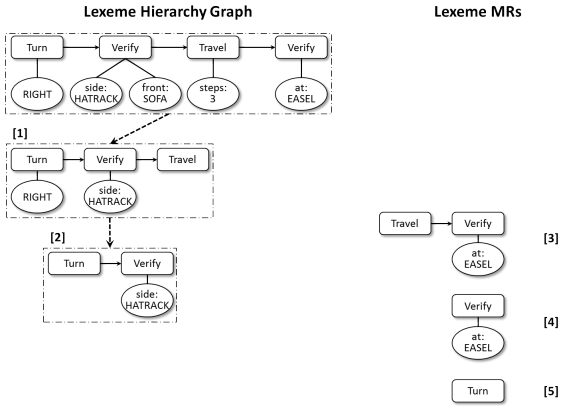
4.3.1 Constructing a Lexeme Hierarchy Graph

An LHG represents the hierarchy of semantic concepts relevant to a particular training instance by encoding the subgraph relations between the MRs of relevant lexemes. Algorithm 1 shows pseudocode of LHG construction for the training instance (e_i, c_i). First, we obtain all relevant lexemes (w_j^i, m_j^i) in the lexicon L, where the MR m_j^i is a subgraph of the context c_i (denoted as $m_j^i \subset c_i$). These lexemes are sorted in descending order based on their MR sizes (i.e. number of nodes in m_j^i). Next, lexemes are inserted, in order, into the MR hierarchy graph starting with the root node of the context c_i. The MR of an added child should be a subgraph of the MR of its parent. Figure 7 illustrates a sample construction of an LHG.

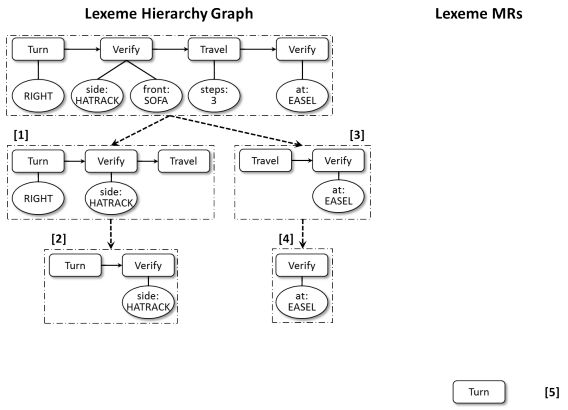
The initial LHG may contain nodes with too many children, which may result in too many PCFG rules because we add a PCFG production rule for every possible k-permutation of the children of each node (see Section 4.3.2). Thus, we introduce *pseudo-lexeme* nodes to reduce the branching factor by repeatedly



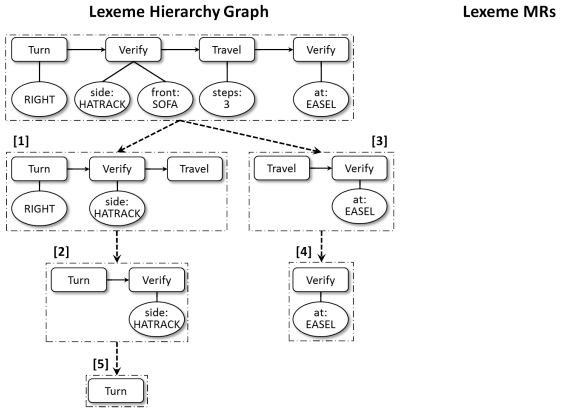
(a) All relevant lexemes are obtained for the training example and ordered by the number of nodes in their MR.



(b) Lexeme MR [1] is added as a child of the top node. MR [2] is a subgraph of [1], so it is added as its child.



(c) MR [3] is not a subgraph of [1] or [2], so it is added as a child of the root. MR [4] is added under [3].



(d) Finally, MR [5] is recursively filtered down and found its right place under [2].

Figure 7: Sample LHG construction for the context `Turn(RIGHT)`, `Verify(side : HATRACK, front : SOFA)`, `Travel(steps : 3)`, `Verify(at : EASEL)`.

Algorithm 2 ADDING PSEUDO LEXEMES TO LHG

Input: LHG with root T
Output: LHG with pseudo lexemes added
procedure RECONSTRUCTLHG(T)
  repeat
    $((T_i, T_j), m') \leftarrow$ MOSTSIMILARPAIR(T)
    Add child T' of T; $MR(T') \leftarrow m'$
    Move T_i and T_j to be children of T'
  until There are no more pairs to combine
  for all non-leaf children T_k of T **do**
    RECONSTRUCTLHG(T_k)
  end for
end procedure

procedure MOSTSIMILARPAIR(T)
  for all pairs (T_i, T_j) of children of T **do**
    $m' \leftarrow$ smallest graph s.t. $MR(T_i) \subset m'$,
      $MR(T_j) \subset m'$, $m' \subset MR(T)$
    $score \leftarrow Sim(MR(T_i), MR(T_j), m')$
    if $maxScore < score$ **then**
      $maxPair \leftarrow (T_i, T_j)$
      $maxScore \leftarrow score$
    end if
  end for
  return $(maxPair, m')$
end procedure

combining the two most similar children of each node. Pseudocode for this procedure is shown in Algorithm 2. The MR for a pseudo-lexeme is the minimal graph, m', which is a supergraph of both of the lexeme MRs that it combines. The pair of most similar children, (m_i, m_j), is calculated by the ratio of how many nodes in m_i and m_j overlap with m' and given by:

$$Sim(m_i, m_j, m') = \frac{|m_i| + |m_j|}{2|m'|}$$

where $|m|$ is the number of nodes in the MR m. Adding pseudo-lexemes has another advantage. They can be intuitively thought of as higher-level semantic concepts composed of two or more concepts. Moreover, the pseudo-lexemes will likely occur in other training examples as well, allowing for more flexible interpretations. For example, let us consider the rule $A \Rightarrow BCD$ from a LHG, and we introduce pseudo lexeme E so that we build two rules $A \Rightarrow BE$ and $E \Rightarrow CD$. It is likely that E occurs in another rule in other training examples such as $E \Rightarrow FG$. Then, we can increase the model's expressiveness by having rules such as $A \Rightarrow^* BFG$, providing more flexibility when parsing a novel NL sentence.

$Root \rightarrow S_c, \quad \forall c \in contexts$

$\forall non\text{-}leaf\ node\ and\ its\ MR\ m$
$S_m \rightarrow \{S_{m_1}, \ldots, S_{m_n}\},$
$where\ m_1, \ldots, m_n: children\ lexeme\ MR\ of\ m,$
$\{\cdot\}: all\ k\text{-}permutations\ for\ k = 1, \ldots, n$

$\forall lexeme\ MR\ m$
$S_m \rightarrow Phrase_m$
$Phrase_m \rightarrow Word_m$ $\quad$ $PhX_m \rightarrow PhX_m\ Word_m$
$Phrase_m \rightarrow PhX_m\ Word_m$ $\quad$ $PhX_m \rightarrow PhX_m\ Word_\emptyset$
$Phrase_m \rightarrow Ph_m\ Word_\emptyset$ $\quad$ $Ph_m \rightarrow PhX_m\ Word_m$
$PhX_m \rightarrow Word_m$ $\quad$ $Ph_m \rightarrow Ph_m\ Word_\emptyset$
$PhX_m \rightarrow Word_\emptyset$ $\quad$ $Ph_m \rightarrow Word_m$
$Word_m \rightarrow s, \quad \forall s \text{ s.t. } (s, m) \in lexicon\ L$
$Word_m \rightarrow w, \quad \forall word\ w \in s \text{ s.t. } (s, m) \in lexicon\ L$

$Word_\emptyset \rightarrow w, \quad \forall word\ w \in NLs$

Figure 8: Summary of the rule generation process. *NLs* refer to the set of NL words in the corpus. Lexeme MR rules follow the schemata of Börschinger et al. (2011), and allow every lexeme MR to generate at least one NL words through an unigram Markov process. Note that pseudo-lexeme nodes do not produce NL words.

4.3.2 Composing PCFG Rules

The next step is to compose PCFG rules from the LHGs and the process is summarized in Figure 8. We basically follow the scheme of Börschinger et al. (2011), but instead of generating NL words from each atomic MR, words are generated from each lexeme MR, and smaller lexeme MRs are generated from more complex ones as given by the LHGs. A nonterminal S_m is generated for the MR, m, of each LHG node. Then, for every LHG node, T, with MR, m, we add rules of the form $S_m \rightarrow S_{m_i} \ldots S_{m_j}$, where the RHS is some k-permutation of the nonterminals for the MRs of the children of node T. Although Börschinger et al. made sure every MR constituent generates at least one NL word, we must generate every possible ordered subset of the children nonterminals, because we do not know which subgraph of the whole context c_i is responsible for generating the NL words in the sentence. In summary, complex semantic concepts are described as an ordered list of smaller concepts which are eventually described by NL phrases.

The rest of the process more closely follows Börschinger et al.'s. Every lexeme MR, m, [3] generates a rule $S_m \rightarrow Phrase_m$, and every $Phrase_m$ generates a sequence of NL words, including one or more "content words" ($Word_m$) for expressing m and zero or more "extraneous" words ($Word_\emptyset$). While Börschinger et al. let $Word_m$ generate any NL words in the vocabulary weighted by EM, we restrict each $Word_m$ to only produce the NL phrases or words associated with m in the lexicon. This helps reduce the PCFG to the tractable size and also decreases unnecessary ambiguity caused by the possible connections between lexemes and all words in the vocabulary. $Word_\emptyset$ has rules for every word including unknown ones, thus it is responsible for generating uncovered words.

[3] Pseudo-lexemes only generate words by generating child lexemes.

Algorithm 3 CONSTRUCT PARSED MR RESULT

Input: Parse tree T for input NL, e, with all $Phrase_x$ subtrees removed.
Output: Semantic parse MR, m, for e
procedure OBTAINPARSEDOUTPUT(T)
  if T is a leaf **then**
    return $MR(T)$ with all its nodes marked
  end if
  for all children T_i of T **do**
    $m_i \leftarrow$ OBTAINPARSEDOUTPUT(T_i)
    Mark the nodes in $MR(T)$ corresponding
      to the marked nodes in m_i
  end for
  if T is not the root **then**
    return $MR(T)$
  end if
  return $MR(T)$ with unmarked nodes removed
end procedure

4.3.3 Parsing Novel NL Sentences

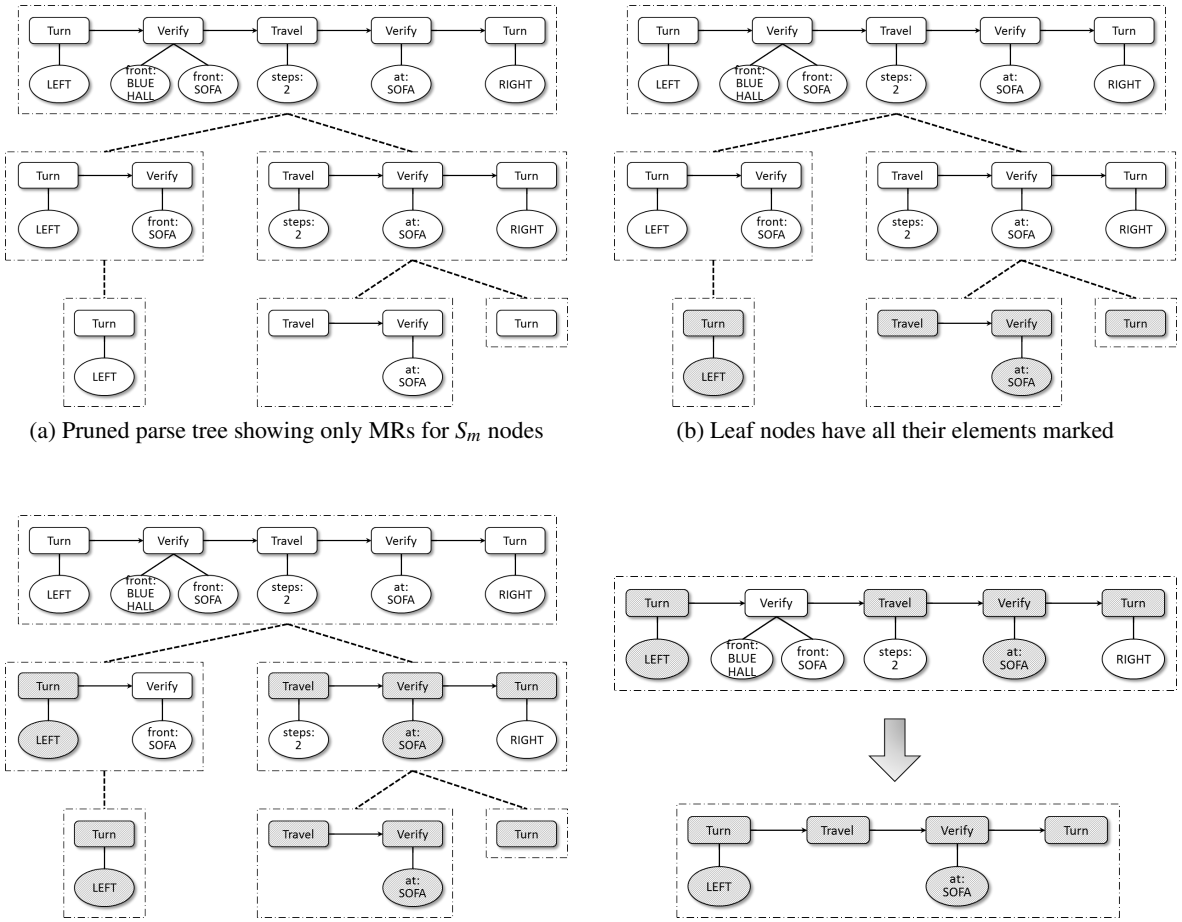
To learn the parameters of the resulting PCFG, we use the Inside-Outside algorithm.[4] Then, the standard probabilistic CKY algorithm is used to produce the most probable parses for novel NL sentences (Jurafsky & Martin, 2000).

Börschinger et al. (2011) simply read the MR, m, for a sentence off the top S_m nonterminal of the most probable parse tree. Therefore, their model is able to produce only the MRs seen during training. Instead, our method produces the output MR parse by composing the appropriate subset of lexeme MRs that are actually responsible for generating NL words. Thus, our system is able to produce novel MRs as long as they are some subgraphs of the complete context (c_i) that appeared in the training data.

First, the parse tree is pruned to remove all the subtrees with the root of $Phrase_x$, producing the tree with only S_m nodes. The pruned subtrees are only concerned about NL generation, so we can figure out which lexeme MRs are involved in generating the target NL sentence. The leaves S_m in the pruned tree show lexeme MRs m that are responsible for generating the NL sentence. These lexeme MR components are combined with respect to the parse tree structure to produce the final MR parse.

Algorithm 3 shows the pseudocode for producing the MR parse from the pruned parse tree. Figure 9 is a sample trace. The algorithm recursively traverses the parse tree. When a leaf-node is reached, it marks all of the nodes in its MR. After traversing all of its children, a node in the MR for the current parse-tree node is marked if and only if its corresponding node in any of the children's MRs were marked. Removing all of the unmarked nodes from the root MR results in the final MR we want.

[4] We used the implementation available at `http://web.science.mq.edu.au/~mjohnson/Software.htm` which was also used by Börschinger et al. (2011).



(a) Pruned parse tree showing only MRs for S_m nodes

(b) Leaf nodes have all their elements marked

(c) Upper level nodes are marked according to leaf-node markings

(d) Removing all unmarked elements for the root node leads to the final MR output

Figure 9: Sample construction of a MR output from a pruned parse tree.

4.4 Experimental Evaluation

4.4.1 Data

We used the English instructions and follower data collected by MacMahon et al.(MacMahon, Stankiewicz, & Kuipers, 2006).[5] This data contains 706 route instructions for three virtual worlds. The instructions were produced by six instructors for 126 unique starting and ending location pairs in the three worlds, and each instruction comes with 1 to 15 human followers traces with an average of 10.4 actions per instruction. Each instruction consists of an average of 5.0 sentences, each containing an average of 7.8 words. Chen and Mooney (2011) constructed the additional single-sentence corpus by matching each sentence with the majority of human followers' actions. This single-sentence version is used for training our model, but both versions are used for testing. There are manually annotated "gold standard" plans only for evaluation purposes.

[5]Data and relevant code are available at `http://www.cs.utexas.edu/users/ml/clamp/navigation/`

	Precision	Recall	F1
Our PCFG model	87.58	∗65.41	∗**74.81**
Chen and Mooney (2011)	∗90.22	55.10	68.37

Table 6: Test accuracy for semantic parsing. '∗' denotes statistically significant difference.

	Single-sentence	Paragraph
Our PCFG model	∗**57.22%**	∗**20.17%**
Chen and Mooney (2011)	54.40%	16.18%

Table 7: Successful plan execution rates using the MARCO execution module on test data. '∗' denotes statistically significant difference.

4.4.2 Methodology and Results

For evaluation, we followed the same methodology as Chen and Mooney (2011), performing "leave one environment out" cross-validation (i.e. training on two environments and testing on the third). We present direct comparison with the best reported results in Chen and Mooney (2011). A Wilcoxon signed-rank test is performed for statistical significance, and '∗' denotes significant differences ($p < .01$) in the tables.

Semantic Parsing Results Semantic parsing evaluates how accurately the model learns to map novel NL sentences in the test environment into correct MRs. Partial semantic-parsing accuracy (Chen & Mooney, 2011) assigns partial credit if two MRs have the same predicate, and additional credit for each matching argument. Precision (accuracy of the system output against the gold standard), recall (the gold standard against the system output), and F1 (harmonic mean between precision and recall) are evaluated and every metric considers partial credit for approximately correct MRs.

Table 6 demonstrates that our method is better than Chen and Mooney's by 6 points in F1. Our PCFG-based approach with semantic lexicon is able to probabilistically disambiguate the training data as well as simultaneously learn a statistical semantic parser within a single framework. This results in better overall performance compared to Chen and Mooney (2011), since they lose possibly useful information due to separate stages of the system, particularly during the refinement stage. In addition, their refinement process is limited to only incorporating the high-scoring lexemes. In contrast, our approach probabilistically considers relatively low score but useful lexemes as well in the generative process, and therefore has more flexibility in the final MR interpretation. This is reflected in the increase of recall for our approach since our method has wider coverage of lexemes during training.

Navigation Plan Execution Results The next evaluation is to test the end-to-end execution of the parsed navigation plans for test instructions in novel environments to see if they reach the exact desired destinations in the environment. Table 7 shows the successful end-to-end navigation task completion rates for both single-sentences and complete paragraph instructions. Following (Chen & Mooney, 2011), the execution is performed by the MARCO (MacMahon et al., 2006) system with the parsed navigation plans output from our model. For the single-sentence corpus, we also considered whether the virtual agent is facing the correct direction compared to the gold-standard.

Our system outperforms Chen and Mooney's best results since more accurate semantic parsing produces more successful plans. However, the difference in performance is smaller than that observed for semantic

parsing. This is because redundancy in human generated instructions allows an incorrect semantic parse to be successful, as long as the errors do not affect its ability to guide the system to the correct destination.

4.5 Discussions

Our approach is novel compared to Börschinger et al. (2011) in the following ways:

- The basic building blocks for associating NL and MR are semantic lexemes instead of atomic MR constituents. This prevents the number of produced PCFG rules from exploding which happens easily in Börschinger et al. (2011) for even a moderately complex MR language. As mentioned earlier, intuitively the lexemes are analogous to syntactic categories in syntax parsing in that complex lexeme MRs represent complicated semantic concepts whereas higher-level syntactic categories such as S, VP, or NP represent complex syntactic structure.
- Our approach has the ability to produce a previously unseen MR, whereas Börschinger et al. (2011) can only generate a parsed MR only if it is included in the PCFG rules constructed from the training data. Even though our MR parse is restricted to be a subset of the training contexts c_is, our model allows for exponentially many combinations.

In addition, our approach also covers wider selections of MR outputs than Chen and Mooney (2011) even though we use their semantic lexicon as our input. Chen and Mooney's system deterministically builds supervised training set by greedily selecting high-score lexemes, thus including only high-score lexemes during the training phase. On the other hand, our probabilistic approach also considers relatively low-score but useful lexemes, therefore covering more semantic concepts in the lexicon. This explains why our approach performs particularly better in recall of the semantic parsing evaluations. Intuitively, we do a better job of utilizing all the semantic lexemes.

Even though we have demonstrated our approach for a fairly specific task, the navigation task, we can apply it to other language grounding tasks as well where a NL sentence is potentially connected to some world states/events/actions expressed as a sequence/set of logical forms. Our approach with LHG provides a general PCFG framework for grounded language learning as long as an appropriate lexicon is provided, since the lexicon learning algorithm can be replaced for other domains.

4.6 Online Semantic Lexicon Learning

Our PCFG induction model is greatly affected by the quality of the semantic lexicon, since semantic lexemes are the basic building blocks for our model. It is interesting to see if different lexicon learning algorithms would increase the overall performance of our model. In this section, I discuss a fast online lexicon learning algorithm called Subgraph Generation Online Lexicon Learning (SGOLL) recently proposed by Chen (2012) and how it will affect our model's performance in the evaluations. Chen demonstrated that SGOLL is relatively effective compared with the system by Chen and Mooney (2011) on the navigation task, while the learning process is much faster.

The algorithm by Chen and Mooney (2011) obtains candidate lexeme MRs for a NL phrase w by repeatedly taking intersections between MRs in the candidate lexeme MR set. Even though it is quite effective for getting a *maximal* meaning for a phrase w, the learning process is slow. SGOLL is inspired by the fact that most words or short phrases correspond to small MR graphs, thus the algorithm focuses only on candidate meanings smaller than a certain size. The process collects co-occurrence information between n-grams w_j and connected subgraphs up to a certain size (in their paper, 3). Since each training example

	Precision	Recall	F1
Our model with Chen and Mooney (2011)	87.58	∗65.41	∗**74.81**
Our model with SGOLL (Chen, 2012)	∗89.04	61.06	72.30

Table 8: Semantic parsing results comparing models using different lexicon. '∗' denotes statistically significant difference.

	Single-sentence	Paragraph
Our model with Chen and Mooney (2011)	∗**57.22%**	∗**20.17%**
Our model with SGOLL (Chen, 2012)	55.01%	18.56%

Table 9: Plan execution rates using the MARCO execution module comparing models with different lexicon. '∗' denotes statistical significance.

is only processed once, SGOLL results in a much faster learning process. The score of candidate lexemes are calculated by the same scoring function as in Chen and Mooney (2011) and the final lexicon output is obtained by ranking the candidate lexemes with the scores.

The experimental results of using SGOLL with our PCFG induction model are shown in Table 8 and Table 9. Wilcoxon signed-rank test is performed for statistical significance and the significance is marked with $*$ ($p < .01$).

The results show that SGOLL does not improve the performance of our model either in semantic parsing or the subsequent plan execution. The primary reason is that since SGOLL is able to only consider small-sized lexemes, the entire LHG structure mainly comes from composing pseudo-lexemes between SGOLL lexemes. This means that LHG with SGOLL may deviate from the real underlying semantics for composite NL phrases, thus showing worse performance overall.

5 Proposed Research

In this section, I will describe some of the short-term proposed work extending the current PCFG induction model explained in Section 4 and also further discuss a long-term agenda towards my thesis. For the short-term work, I will first discuss employing better learning algorithms for acquiring a semantic lexicon using part-of-speech (POS) tags, since our model relies on the quality of the semantic lexicon input. Next, I will describe how I could improve the produced MR parses from our PCFG model by using discriminative re-ranking algorithms. With plan execution feedback and other general features, we can train an averaged perceptron algorithm (Collins, 2002a) to rank the top N parse outputs. Moreover, I plan to investigate how the *meaning representation grammar* (MRG) can help the current PCFG model. Currently, the MRs are considered as the congregation of atomic constituents, but further structural information from the MRL grammar which defines how MRs are generated will provide additional useful cues to improve semantic correspondence with NL segments.

The long-term goals include application of the current approach to the area of machine translation, extending the ambiguous semantic parsing model to deal with real perception data such as images/videos and sensory data, and co-learning with external knowledge such as information extraction, syntax parsing, or entity recognition.

5.1 Improved Semantic Lexicon Learning with Part-of-Speech Tags

Our PCFG induction model presented in Section 4 is highly affected by the quality of the semantic lexicon. If the input lexicon is noisy, the model will make poor choices connecting NL words to lexeme MRs, thus degrading the accuracy of the final MR parses. If we employ a cleaner semantic lexicon, we can expect better performance for the entire model in the evaluations. In this section, I discuss a possible improvement of the lexicon learning algorithm from Chen and Mooney (2011) by using part-of-speech (POS) tags. However, there are a number of possibilities for improving the semantic lexicon learning algorithms used in our PCFG model. In general, our approach with LHG provides a general PCFG framework for grounded language learning with the appropriate lexicon for the target domains.

Guo and Mooney(unpublished) recently proposed a simple approach of enhancing the quality of the semantic lexicon by using part-of-speech (POS) tags. The idea behind this is called "syntactic bootstrapping", which means that the meanings of natural language can be aided by previously acquired syntactic knowledge. A several piece of research in psycholinguistics showed that the process of learning the meanings of novel words by children is benefited by such syntactic knowledge (Gleitman, 1990; Gleitman & Landau, 1994). Guo and Mooney's approach is based on the same navigation task framework (Chen & Mooney, 2011) that our PCFG induction model is evaluated on. This approach enhances the quality of the semantic lexicon learning algorithm of Chen (2012) by constraining verbs to map to `Action`*s* and nouns to `Object`*s* in the lexicon learning process. Even though their method is performed on the work of Chen (2012), we can apply the same idea to refine the lexicon created by Chen and Mooney (2011).

The lexicon learning algorithm of Chen and Mooney (2011) is purely correlational which is a common method for learning a semantic lexicon (Thompson & Mooney, 2003). However, such correlational methods sometimes produce incorrect lexicon entries because correlation between an *n*-gram *w* and a MR *m* can be misinterpreted when *w* is typically used in certain contexts. Thus, if we disallow some of the learned lexemes when there is a violation to the rule of connecting verbs to `Action`*s* and nouns to `Object`*s*, such inaccuracies of correlational methods will be alleviated.

The process first starts with the initial learned semantic lexicon. Every word in the NL phrase of every lexeme is tagged with POS tags by the Stanford POS tagger (Toutanova, Klein, Manning, & Singer, 2003) trained on the Wall Street Journal treebank to POS-tag English. Then, each element of MR in the lexeme is labeled: `Turn`/`Travel`/`Verify` are marked as `Actions`, and other arguments are marked as `Objects`. Finally, if a lexeme violates the following constraints, it is simply removed from the semantic lexicon:

- *n*-gram *w* contains a noun if and only if MR *m* contains an `Object`.
- *n*-gram *w* contains a verb if and only if MR *m* contains an `Action`.

The experimental results of this enhancement on the lexicon learning algorithm show modest improvement over the default lexicon learning algorithm under the same system of Chen (2012). Similarly, our PCFG induction model can simply use this improved semantic lexicon instead. However, the refinement step uses the prior linguistic knowledge obtained through a POS tagger (Biemann, 2006; Ravi & Knight, 2009; Goldwater & Griffiths, 2007) trained on a WSJ corpus, which exploits unfair advantage. Therefore, what I propose is to learn unsupervised POS tagging from the navigation data and help improve the semantic lexicon learned by the algorithm of Chen and Mooney (2011). In addition, the current refinement method using POS tagging assumes prior knowledge that verbs match to `Actions` and nouns match to `Objects`. However, there are other possibly useful relations that we can exploit further. For instance, `Turn` or `Travel` actions tend to be related to verbs mostly, whereas `Verify` actions are more related to descriptive propositional phrases. This is because we do not usually say things like "go 2 blocks and check you are at the

sofa", but rather we say "go 2 blocks until you are at the sofa". For another example, `Turn` actions are more likely to be linked with nouns, since it is common to give instructions like "make a right turn". Such specific relations cannot be encoded fully manually, because these require domain-specific knowledge. I propose to learn such detailed relations through joint learning of unsupervised POS tagging and semantic lexicon, and this joint process not only could improve the quality of the semantic lexicon but could also enhance POS tagging for the domain.

5.2 Discriminative Re-ranking of Parsed Results

Discriminative re-ranking is a common machine learning technique to improve the output of generative models. In re-ranking methods, a baseline generative model is trained and used to generate a set of candidate outputs for each training example. Then, a second conditional model incorporates more complex features than the baseline model and is used to re-rank the candidates, exploring the importance of global features to produce better final outputs (Collins, 2000). Re-ranking has been shown to be effective in various natural language processing tasks including many tagging and parsing tasks (Collins, 2000, 2002a, 2002c, 2002b; Collins & Koo, 2005; Charniak & Johnson, 2005) along with semantic role labeling (Toutanova, Haghighi, & Manning, 2005). In addition, Ge and Mooney (2006) and Lu et al. (2009) also showed that discriminative re-ranking help enhance the quality of semantic parsers.

The MR output of our PCFG induction model can also be improved via discriminative re-ranking. Although our model deals with semantic parsing from ambiguous training, the model itself has the form of standard PCFG induction where discriminative re-ranking has been shown to be effective in prior work.

5.2.1 Averaged Perceptron Algorithm for Re-ranking

The averaged perceptron algorithm (Collins, 2002a) has been successfully applied to various natural language processing tasks including re-ranking for tagging and parsing. In this section, I will describe how the averaged perceptron algorithm can be applied to improve the parsed output of our PCFG model. The algorithm requires three subcomponents: a GEN function defining a set of top k candidate parse trees for each NL sentence, a feature function Φ that maps a NL sentence, e, and a parse tree, y, into a feature vector such that $\Phi(e,y) \in R^d$, an evaluation function that executes the MR output from a candidate parse tree and estimates how well it reaches the intended correct destination. The third component should normally be the reference parse tree in supervised semantic parsing tasks. However, we do not know the gold-standard parse trees in the navigation domain and thus we need to use the indirect supervision of execution responses instead, measuring how much of the correct route is successfully reached by the candidate MR plan. In this sense, our re-ranking approach is different from other general discriminative re-ranking methods, since we do not have the gold-standard reference tree. Instead, we introduce pseudo gold-standard parse tree in place of the reference gold-standard tree. It is selected among the top k candidate parse trees, whose derived MR plan achieves the best execution score out of all. Even though this does not guarantee the 100% correct MR plan to be referenced, the candidate parse trees can be ranked in the order of maximizing the execution accuracy, which is desirable for the goal of the navigation task.

The algorithm learns a weight vector W that maps a weight to each feature so that the score $W \cdot \Phi(e,y)$ is assigned to each candidate parse tree for each NL sentence e. Then, the trained perceptron produces the final parse output with the highest score when given a new NL sentence. The algorithm is briefly described in Algorithm 4.

Algorithm 4 PERCEPTRON TRAINING ALGORITHM

Input: A set of training examples (e_i, y_i^*), where e_i is a NL sentence, y_i^* is a candidate parse tree whose composed MR output m_i^* has the highest execution evaluation score out of all the candidate parses.
Output: The parameter vector $\bar{W}$
procedure PERCEPTRON
  Initialize $\bar{W} = 0$
  for do$t = 1...T, i = 1...n$
    Obtain $y_i = \arg\max_{y \in GEN(e_i)} \Phi(e_i, y) \cdot \bar{W}$
    if $y_i \neq y_i^*$ **then**
      $\bar{W} = \bar{W} + \Phi(e_i, y_i^*) - \Phi(e_i, y_i)$
    end if
  end for
end procedure

5.2.2 Features

The features we plan to use for training the averaged perceptron algorithm are the following:

1. PCFG Rule. Indicate whether a PCFG rule is used in the parse tree y such as $A \rightarrow BC$.

2. Grandparent PCFG Rule. Indicate whether a PCFG rule *a*s well as the nonterminal above that is used in y. For instance, if there are rules $A \rightarrow BC$ and $C \rightarrow DE$, then the grandparent PCFG rule feature will be $I(C \rightarrow DE, A)$, where $I(\cdot)$ is indicator function.

3. Long-range Unigram Rule. Indicate whether a nonterminal has a given NL word below it in the parse tree. If a nonterminal A eventually produces a NL word w, then the feature value of the long-range unigram rule becomes 1.

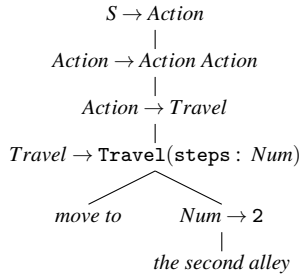
4. Two-level Unigram Rule. Indicate whether a PCFG production has a child nonterminal which eventually generates a NL word in the parse tree. For instance, for the rule $A \rightarrow BC$, if B or C has a particular NL word below it in the parse tree, then the feature value is 1.

5. Log-probability of Parse Tree. Include the log value of the joint probability of the candidate parse tree which can be obtained from the product of all the production rules used in the parse tree.

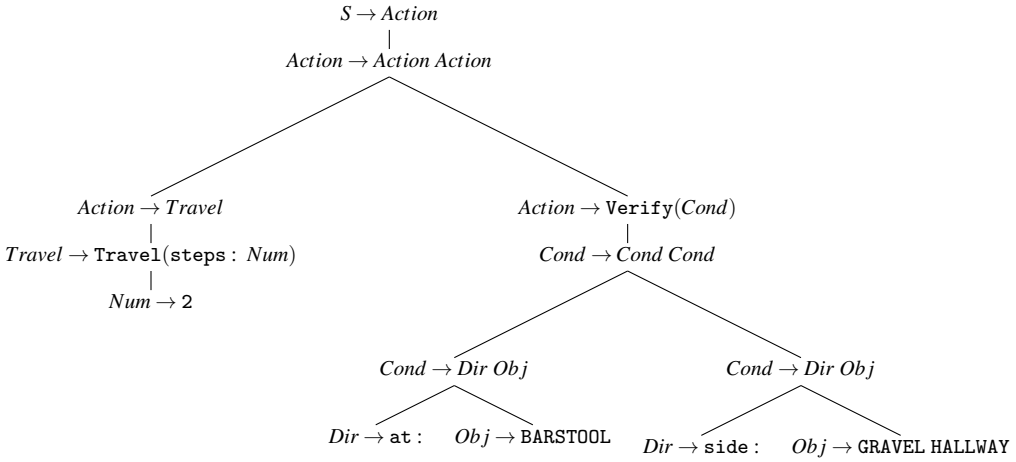
The above features are adapted mostly from Collins and Koo (2005) and Lu et al. (2009). Features 1-4 are indicator functions taking value 1 if a designated combination exists in the parse tree and value 0 otherwise. Feature 5 is a real-valued function that measures how likely a candidate parse tree is based on the trained generative model.

5.3 Incorporating MRL Grammar Structure

Our PCFG induction model treats MRs as mere compositions of atomic constituents. This means that our model only focuses on *what* element is generated rather than *how* the element is generated. Since the meaning representation language (MRL) has its own grammar rules defining how the MR structure is constructed, our PCFG model could utilize these additional useful information to learn the correspondences between NL and MR.



(a) Simplified PCFG parse tree which the proposed model will produce. Note that only the grammar rules which actually generate NL words are included in the tree.



(b) MRL grammar structure of the context.

Figure 10: Sample simplified PCFG parse tree constructed based on MR grammar rule structure and full MR structure, where the instruction/context pair is *move to the second alley* / Travel(steps : 2), Verify(at : BARSTOOL, side : GRAVEL HALLWAY).

There have been several attempts at learning a semantic parser that takes MRL grammar into consideration. KRISP (Kernel-based Robust Interpretation for Semantic Parsing) (Kate & Mooney, 2006) is a semantic parser learner that uses support vector machines (SVMs) with string kernels. For each production rule in the MRL grammar, KRISP learns an SVM string classifier to recognize a particular word or phrase. Then, the trained set of classifiers is used to compose the most probable MR for a novel NL sentence. KRISP has been shown experimentally to be particularly robust to noisy training data with partial matching of substrings by string kernels and overfitting prevention by SVMs. WASP (Wong & Mooney, 2006) is a semantic parser learner based on statistical machine translation techniques. It induces a probabilistic synchronous context-free grammar (PSCFG) (Wu, 1997) which simultaneously produces NL and the corresponding MR using the MRL grammar. Statistical bilingual word alignment algorithm GIZA++ (Och & Ney, 2003; Brown et al., 1993) is first used for building a bilingual lexicon between NL words and MR grammar rules which is then used to construct SCFG rules. A maximum entropy model is trained for SCFG rule parameters to produce a semantic parser. In addition, the generative Hybrid Tree model (Lu et al., 2008; Kim & Mooney, 2010) incorporates the MR grammar rules in its generative process to produce the subsequent MR components and NL words. MR grammar rules are responsible for generating children MRs' semantic categories as well as relevant NL words in a single tree structure, and parameters are learned using EM to produce a statistical semantic parser. All these methods construct a compositional structure involving both MRL grammar rules and NL words for learning semantic parsers.

Inspired by these approaches, our PCFG induction model can be modified to construct a hierarchical structure using the MRL grammar instead of the LHG. Using lexemes as the basic building blocks has an advantage in that each lexeme MR works as a complete unit to be used later to form a novel parsed MR output. However, sometimes there are unnecessarily large lexeme MRs that are noisy inputs to the model. Instead, if we use each MR production rule as a building block to be connected to NL words, then fine-grained levels of meanings will be captured, thus resulting in more expressiveness for the model. This proposed model will first build the hierarchy structure based on the MRL rules for the context c_i in the training data, and then a PCFG production rule will be generated for each node in the hierarchy so that each

MRL rule produces some ordered subset of children MRL rules or NL words. Unlike the prior approaches which are fully supervised models and take all of the MR production rules into consideration to match NL words, this proposed model will probabilistically choose which MRL rules are useful for generating NL sentences after being optimized by EM. Figure 10 shows a sample PCFG parse tree which the proposed model will build from an example in the navigation dataset.

5.4 Long-term Directions

In this section, I will discuss some of the longer term directions that I plan to pursue in the future beyond the scope of my thesis.

5.4.1 Joint Learning with Other NLP Tasks

The completed works discussed earlier do not incorporate preprocessing of natural language texts before we learn to find probabilistic mappings to MR components. This also means that we do not make any prior assumptions about the language itself and we have to learn the underlying meanings of the raw NL segments from scratch in a statistical manner. However, we also know that some stop words such as auxiliary verbs do not play an important role in understanding the semantics of language. Moreover, additional information such as part-of-speech tags, semantic role labels, or syntactic parse trees will help bias the generation process from MR elements to natural language phrases.

For instance, POS-tagged natural language words will have limited choices for which lexeme MRs or atomic MR constituents they can be matched to. Guo and Mooney(unpublished) used POS tags only for filtering out inappropriate lexicon entries. However, POS tags can also be integrated in the generative process itself within the PCFG framework so that lexeme MRs or MR grammar rules first generate appropriate POS tags and then subsequently produce the corresponding NL words. This process will jointly learn unsupervised POS tags as well as the PCFG structure that connects MR components/rules to NL words. In addition, information extraction techniques can help identify notably important NL words which are actually meaningful and help reduce the ambiguity in selecting correct subparts out of the entire ambiguous MRs through a similar integration. Finally, syntactic parse trees are another option to consider which can be integrated into the CFG trees produced by our PCFG model as Ge and Mooney (2006) did. As mentioned earlier, syntactic hierarchies intuitively resemble an LHG, and it is possible to combine both structures to utilize both cues for an improved model.

5.4.2 Machine Translation

Semantic parsing is a particular form of machine translation from a source natural language to a target meaning representation language. WASP (Wong & Mooney, 2006) is inspired by this notion and builds a semantic parsing model based on state-of-the-art syntax-based statistical machine translation (SMT) (Chiang, 2005). In addition, it is also noteworthy that many existing semantic parsing approaches implicitly use structural similarity between NL and MR. In this sense, it is would be interesting to apply our semantic parsing methods back to the SMT tasks. Conventional SMT tasks resemble conventional supervised semantic parsing tasks, since the conventional SMT approaches require a sentence aligned parallel corpus to train the models. Our navigation task with highly ambiguous supervision is analogous to the task of "summarized translation". In the navigation task, our PCFG model finds the probabilistic correspondences between the semantic concepts represented by lexeme MRs and the NL phrases, where only some subparts of the given context (the full landmarks plan) are referred to by the given NL sentence. In the summarized translation

task, a rich text in a source language is translated to a more concise text in a target language containing only the gist of the original text. For example, many Wikipedia articles are published in many languages, but the contents are usually not parallel between languages. Contents in some languages are rich in details, but contents in other languages may contain only the gist of the topic. Currently, such summarized translation has not been investigated to our knowledge. However, beyond SMT trained on parallel corpora, there have been various attempts at using "comparable corpora" which are collections of documents that are comparable and similar in content and form in various degrees and dimensions (Munteanu & Marcu, 2005; Diab & Finch, 2000; Snover, Dorr, & Schwartz, 2008). However, the documents in comparable corpora usually have similar levels of complexity and thus the task has very different characteristics from summarized translation. Summarized translation can be thought of as the ambiguous correspondence problem between two languages. Only some subparts of the rich language text are translated to the concise target language. Based on the idea of our PCFG induction model that finds rich-to-concise correspondences, we could extend and apply our model to the summarized translation task in conjunction with SMT techniques.

5.4.3 Real Perceptional Data

In the completed work presented in Section 3 and 4, we assumed for simplicity that we have abstracted information of the surrounding world states which is ambiguously connected to natural language sentences. An obvious extension of our model would be to learn directly from real perceptional data instead of abstracted logical representations. However, we first need a way to select what raw information is important and notable to be learned with natural language. A recent work (Chao, Cakmak, & Thomaz, 2011) investigated how to train a socially interactive robot through demonstration. They grounded the meaning of natural language by the state changes that appeared in the sensor readings. Even though their work limits the tasks to only several categories and the complexity of natural language used is simple, it is notable that they were able to train a machine learning model which directly connects natural language instruction to the changes in real-valued raw sensory data. In a similar way, we can construct a semantic lexicon composed of state changes in sensor data, and then our PCFG induction model can be applied to further investigate NL–MR groundings with proper modifications. In addition, with a provided object recognition system, our presented system can identify surrounding environments as well as notable landmarks to help interpret natural language instructions and follow them to navigate in the real environments as in other navigation work (Shimizu & Haas, 2009; Matuszek et al., 2010; Vogel & Jurafsky, 2010; Kollar et al., 2010; Tellex et al., 2011). Moreover, we can also extend and apply our methods to visual data such as images and videos. Extracted features such as SIFT (Lowe, 1999) and space-time interest points (Laptev, 2005) in images and videos will produce vectors that can be used to describe notable objects or landmarks. Then, our model will automatically discover how certain visual features are probabilistically related to natural language.

6 Conclusion

The ultimate goal of grounded language learning for computational systems is to mimic the process of human language learning. In many cases, it is reduced to the problem of learning the underlying semantics of languages in ambiguous perceptual environments. This type of learning approach has gained growing attention recently, since there is no need for explicit human intervention to acquire training data. By contrast, conventional semantic learning methods such as supervised semantic parsing require costly, hard to acquire, one-to-one full supervision of natural language sentences and the corresponding logical forms.

In this proposal, we presented two grounded language learning approaches dealing with such natural,

ambiguous supervision. The RoboCup sportscasting task and the navigation task are publicly available datasets collected from the automatic extraction of world state information associated with corresponding natural language texts. Prior grounded language learning works on these datasets suffer possible information loss due to a lack of exploiting the probabilistic nature of connections between NL and MR, or only focusing on semantic alignments and not on learning the underlying semantics of languages. In contrast, the two presented completed works use generative processes to select NL and MR components probabilistically, simultaneously learning semantic alignment and meanings of natural languages.

For the short-term future work, I propose to extend our probabilistic PCFG induction model with semantic lexicons in various ways: improving the lexicon learning algorithms with POS tags, discriminative re-ranking of the top-k parses to improve the final outputs, integrating the MRL grammar structure to achieve better expressiveness. Long term agenda includes application to summarized machine translation using our completed work for ambiguous supervision, application to real perception data such as sensory data and extracted features of images/videos, and joint learning with other NLP tasks which can help bias our models for better prediction.

Acknowledgements

We thank David Chen and Lu Guo for helpful comments and providing source code of their lexicon learning algorithms. We also thank Wei Lu for his work and code for the Hybrid Tree Model. This work was funded by the NSF grant IIS-0712907 and IIS-1016312. Experiments were performed on the Mastodon Cluster, provided by NSF Grant EIA-0303609.

References

Aho, A. V., & Ullman, J. D. (1972). *The Theory of Parsing, Translation, and Compiling*. Prentice Hall, Englewood Cliffs, NJ.

Bailey, D., Feldman, J., Narayanan, S., & Lakoff, G. (1997). Modeling embodied lexical development. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*.

Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D. M., & Jordan, M. I. (2003). Matching words and pictures. *Journal of Machine Learning Research*, *3*, 1107–1135.

Barzilay, R., & Lapata, M. (2005). Collective content selection for concept-to-text generation. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-05)*.

Biemann, C. (2006). Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the 21st International Conference on computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, COLING ACL '06, pp. 7–12 Stroudsburg, PA, USA. Association for Computational Linguistics.

Bordes, A., Usunier, N., & Weston, J. (2010). Label ranking under ambiguous supervision for learning semantic correspondences. In *ICML*, pp. 103–110.

Börschinger, B., Jones, B. K., & Johnson, M. (2011). Reducing grounded learning tasks to grammatical inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pp. 1416–1425 Stroudsburg, PA, USA. Association for Computational Linguistics.

Branavan, S., Chen, H., Zettlemoyer, L. S., & Barzilay, R. (2009). Reinforcement learning for mapping instructions to actions. In *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)* Singapore.

Branavan, S., Zettlemoyer, L., & Barzilay, R. (2010). Reading between the lines: Learning to map high-level instructions to commands. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1268–1277. Association for Computational Linguistics.

Brown, P. F., Della Pietra, V. J., Della Pietra, S. A., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, *19*(2), 263–312.

Carroll, J., Copestake, A., Flickinger, D., & Poznanski, V. (1999). An efficient chart generator for (semi-) lexicalist grammars. In *Proceedings of the 7th European workshop on natural language generation (EWNLG99)*, pp. 86–95.

Carroll, J., & Oepen, S. (2005). High efficiency realization for a wide-coverage unification grammar. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-2005)*, pp. 165–176 Jeju Island, Korea.

Chao, C., Cakmak, M., & Thomaz, A. (2011). Towards grounding concepts for transfer in goal learning from demonstration. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, Vol. 2, pp. 1–6. IEEE.

Charniak, E., & Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pp. 173–180 Ann Arbor, MI.

Chen, D. L. (2012). Fast online lexicon learning for grounded language acquisition. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL-2012)* Jeju, Republic of Korea.

Chen, D. L., Kim, J., & Mooney, R. J. (2010). Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, *37*, 397–435.

Chen, D. L., & Mooney, R. J. (2008). Learning to sportscast: a test of grounded language acquisition. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pp. 128–135.

Chen, D. L., & Mooney, R. J. (2011). Learning to interpret natural language navigation instructions from observations. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-2011)* San Francisco, CA, USA.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pp. 263–270 Ann Arbor, MI.

Clarke, J., Goldwasser, D., Chang, M.-W., & Roth, D. (2010). Driving semantic parsing from the world's response. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pp. 18–27 Uppsala, Sweden. Association for Computational Linguistics.

Collins, M. (2000). Discriminative reranking for natural language parsing. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)*, pp. 175–182 Stanford, CA.

Collins, M. (2002a). Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP-02)* Philadelphia, PA.

Collins, M. (2002b). New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pp. 263–270 Philadelphia, PA.

Collins, M. (2002c). Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pp. 489–496 Philadelphia, PA.

Collins, M., & Koo, T. (2005). Discriminative reranking for natural language parsing. *Computational Linguistics*, *31*(1), 25–69.

Diab, M., & Finch, S. (2000). A statistical word-level translation model for comparable corpora. In *Proceedings of the Conference on Content-based Multimedia Information Access (RIAO)*.

Fleischman, M., & Roy, D. (2007). Situated models of meaning for sports video retrieval. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT-07)* Rochester, NY.

Ge, R., & Mooney, R. J. (2006). Discriminative reranking for semantic parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL-06)* Sydney, Australia.

Gleitman, L., & Landau, B. (Eds.). (1994). *The Acquisition of the Lexicon*. MIT Press, Cambridge, MA.

Gleitman, L. R. (1990). Structural sources of verb meaning. *Language Acquisition*, *1*, 3–55.

Gold, K., & Scassellati, B. (2007). A robot that uses existing vocabulary to infer non-visual word meanings from observation. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*.

Goldwater, S., & Griffiths, T. (2007). A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 744–751 Prague, Czech Republic. Association for Computational Linguistics.

Guo, L., & Mooney, R. J. Using part-of-speech to aid grounded learning of word meanings..

Gupta, S., Kim, J., Grauman, K., & Mooney, R. (2008). Watch, listen & learn: Co-training on captioned images and videos. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD-08)*, pp. 457–472 Antwerp, Belgium.

Gupta, S., & Mooney, R. (2009). Using closed captions to train activity recognizers that improve video retrieval. In *Proceedings of the CVPR-09 Workshop on Visual and Contextual Learning from Annotated Images and Videos (VCL)* Miami, FL.

Jurafsky, D., & Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, NJ.

Kate, R. J., & Mooney, R. J. (2006). Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL-06)*, pp. 913–920 Sydney, Australia.

Kate, R. J., & Mooney, R. J. (2007). Learning language semantics from ambiguous supervision. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, pp. 895–900 Vancouver, Canada.

Kay, M. (1996). Chart generation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, pp. 200–204 San Francisco, CA.

Kim, J., & Mooney, R. J. (2010). Generative alignment and semantic parsing for learning from ambiguous supervision. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 543–551. Association for Computational Linguistics.

Kim, J., & Mooney, R. J. (2012). Unsupervised PCFG induction for grounded language learning with highly ambiguous supervision. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*, EMNLP-CoNLL '12.

Kollar, T., Tellex, S., Roy, D., & Roy, N. (2010). Toward understanding natural language directions. In *Proceedings of Human Robot Interaction Conference (HRI-2010)*.

Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., & Steedman, M. (2010). Inducing probabilistic ccg grammars from logical form with higher-order unification. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pp. 1223–1233. Association for Computational Linguistics.

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*(2-3), 107–123.

Li, J., & Wang, J. Z. (2008). Real-time computerized annotation of pictures. *IEEE Trans. Pattern Anal. Mach. Intell.*, *30*(6), 985–1002.

Li, L.-J., Socher, R., & Fei-Fei, L. (2009). Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*.

Liang, P., Jordan, M. I., & Klein, D. (2009). Learning semantic correspondences with less supervision. In *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)* Singapore.

Liang, P., Jordan, M. I., & Klein, D. (2011). Learning dependency-based compositional semantics. In *Proceedings of ACL* Portland, Oregon. Association for Computational Linguistics.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Vol. 2.

Lu, W., Ng, H. T., & Lee, W. S. (2009). Natural language generation with tree conditional random fields. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 400–409 Morristown, NJ, USA. Association for Computational Linguistics.

Lu, W., Ng, H. T., Lee, W. S., & Zettlemoyer, L. S. (2008). A generative model for parsing natural language to meaning representations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08)* Honolulu, HI.

MacMahon, M., Stankiewicz, B., & Kuipers, B. (2006). Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)* Boston, MA.

Matuszek, C., Fox, D., & Koscher, K. (2010). Following directions using statistical machine translation. In *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*, HRI '10, pp. 251–258 New York, NY, USA. ACM.

Munteanu, D. S., & Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, *31*(4), 477–504.

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, *29*(1), 19–51.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pp. 311–318 Philadelphia, PA.

Ravi, S., & Knight, K. (2009). Minimized models for unsupervised part-of-speech tagging. 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*, ACL '09, pp. 504–512 Stroudsburg, PA, USA. Association for Computational Linguistics.

Roy, D. (2002). Learning visually grounded words and syntax for a scene description task. *Computer Speech and Language*, *16*(3), 353–385.

Saffran, J. (2003). Statistical language learning mechanisms and constraints. *Current directions in psychological science*, *12*(4), 110–114.

Saffran, J., Johnson, E., Aslin, R., & Newport, E. (1999). Statistical learning of tone sequences by human infants and adults. *Cognition*, *70*(1), 27–52.

Shimizu, N., & Haas, A. (2009). Learning to follow navigational route instructions. In *Proceedings of the Twenty First International Joint Conference on Artificial Intelligence (IJCAI-2009)*.

Siskind, J. M. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, *61*(1), 39–91.

Snover, M., Dorr, B., & Schwartz, R. (2008). Language and translation model adaptation using comparable corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pp. 857–866 Stroudsburg, PA, USA. Association for Computational Linguistics.

Snyder, B., & Barzilay, R. (2007). Database-text alignment via structured multilabel classification. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-2007)*.

Tellex, S., Kollar, T., Dickerson, S., Walter, M., Banerjee, A., Teller, S., & Roy, N. (2011). Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Thompson, C. A., & Mooney, R. J. (2003). Acquiring word-meaning mappings for natural language interfaces. *Journal of Artificial Intelligence Research*, *18*, 1–44.

Toutanova, K., Haghighi, A., & Manning, C. D. (2005). Joint learning improves semantic role labeling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pp. 589–596 Ann Arbor, MI.

Toutanova, K., Klein, D., Manning, C., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Human Language Technology Conference (HLT-NAACL 2003)*.

Vogel, A., & Jurafsky, D. (2010). Learning to follow navigational directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*.

Wang, C., Blei, D. M., & Li, F.-F. (2009). Simultaneous image classification and annotation. In *CVPR*, pp. 1903–1910.

White, M. (2006). CCG chart realization from disjunctive inputs. In *Proceedings of the Fourth International Natural Language Generation Conference*, pp. 12–19. Association for Computational Linguistics.

White, M., & Baldridge, J. (2003). Adapting chart realization to CCG. In *Proceedings of the 9th European Workshop on Natural Language Generation*, pp. 119–126.

White, M. (2004). Reining in CCG chart realization. In *Proceedings of the 3rd International Conference on Natural Language Generation (INLG-2004)* New Forest, UK.

Wong, Y., & Mooney, R. J. (2006). Learning for semantic parsing with statistical machine translation. In *Proceedings of Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL-06)*, pp. 439–446 New York City, NY.

Wong, Y., & Mooney, R. J. (2007a). Generation by inverting a semantic parser that uses statistical machine translation. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT-07)*, pp. 172–179 Rochester, NY.

Wong, Y., & Mooney, R. J. (2007b). Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, pp. 960–967 Prague, Czech Republic.

Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, *23*(3), 377–403.

Yu, C., & Ballard, D. H. (2004). On the integration of grounding language and learning objects. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pp. 488–493.

Zaragoza, H., & Li, C.-H. (2005). Learning what to talk about in descriptive games. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-05)*, pp. 291–298 Vancouver, Canada.

Zelle, J. M., & Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 1050–1055 Portland, OR.

Zettlemoyer, L. S., & Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of 21st Conference on Uncertainty in Artificial Intelligence (UAI-2005)* Edinburgh, Scotland.

Zettlemoyer, L. S., & Collins, M. (2007). Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-07)*, pp. 678–687 Prague, Czech Republic.

Zettlemoyer, L. S., & Collins, M. (2009). Learning context-dependent mappings from sentences to logical form. 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*, pp. 976–984. Association for Computational Linguistics.