Unsupervised PCFG Induction for Grounded Language Learning with Highly Ambiguous Supervision

Joohyun Kim **Raymond J. Mooney**
Department of Computer Science
The University of Texas at Austin
1616 Guadalupe, Suite 2.408
Austin, TX 78701, USA
{scimitar,mooney}@cs.utexas.edu

Abstract

"Grounded" language learning employs training data in the form of sentences paired with relevant but ambiguous perceptual contexts. Börschinger et al. (2011) introduced an approach to grounded language learning based on unsupervised PCFG induction. Their approach works well when each sentence potentially refers to one of a small set of possible meanings, such as in the sportscasting task. However, it does not scale to problems with a large set of potential meanings for each sentence, such as the navigation instruction following task studied by Chen and Mooney (2011). This paper presents an enhancement of the PCFG approach that scales to such problems with highly-ambiguous supervision. Experimental results on the navigation task demonstrates the effectiveness of our approach.

1 Introduction

The ultimate goal of "grounded" language learning is to develop computational systems that can acquire language more like a human child. Given only supervision in the form of sentences paired with relevant but ambiguous perceptual contexts, a system should learn to interpret and/or generate language describing situations and events in the world. For example, systems have learned to commentate simulated robot soccer games by learning from sample sportscasts (Chen and Mooney, 2008; Liang et al., 2009; Börschinger et al., 2011), or understand navigation instructions by learning from action traces produced when following the directions (Chen and Mooney, 2011; Tellex et al., 2011).

Börschinger et al. (2011) recently introduced an approach to grounded language learning using unsupervised induction of *probabilistic context free grammars* (PCFGs) to learn from ambiguous contextual supervision. Their approach first constructs a large set of production rules from sentences paired with descriptions of their ambiguous context, and then trains the parameters of this grammar using EM. Parsing a novel sentence with this grammar gives a parse tree which contains the formal *meaning representation* (MR) for this sentence. This approach works quite well on the sportscasting task originally introduced by Chen and Mooney (2008). In this task, each sentence in a natural-language commentary describing activity in a simulated robot soccer game is paired with the small set of actions observed within the past 5 seconds, one of which is usually described by the sentence. Even with this low level of ambiguity in a constrained domain, their method constructs a PCFG with about 33,000 productions. More fundamentally, their approach is restricted to a finite set of potential meaning representations, and the grammar size grows at least linearly with the number of possible MRs, which in turn is inevitably exponential in the number of objects and actions in the domain.

The navigation task studied by Chen and Mooney (2011) provides much more ambiguous supervision. In this task, each instructional sentence is paired with a formal *landmarks plan* (represented as a large graph) that includes a full description of the observed actions and world-states that result when

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 433–444, Jeju Island, Korea, 12–14 July 2012.

someone follows this instruction. An instruction generally refers to a subgraph of this large graph. Therefore, there are a combinatorial number of possible meanings to which a given sentence can refer.

Chen and Mooney (2011) circumvent this combinatorial problem by never explicitly enumerating the exponential number of potential meanings for each sentence. Their system first induces a semantic lexicon that maps words and short phrases to formal representations of actions and objects in the world. This lexicon is learned by finding words and phrases whose occurrence highly correlates with specific observed actions and objects in the simulated environment when executing the corresponding instruction. This learned lexicon is then used to directly infer a formal MR for observed instructional sentences using a greedy covering algorithm. These inferred MRs are then used to train a supervised semantic parser capable of mapping novel sentences to their formal meanings.

We present a novel enhancement of Börschinger et al.'s PCFG approach that uses Chen and Mooney's lexicon learner to avoid a combinatorial explosion in the number of productions. The learned lexicon is first used to build a hierarchy of semantic *lexemes* (i.e. lexicon entries) called the *Lexeme Hierarchy Graph* (LHG) for each ambiguous landmarks plan in the training data. The intuition behind utilizing an LHG is that the MR for each lexeme constitutes a semantic concept that corresponds to some natural-language (NL) word or phrase. Therefore, the LHG represents how complex semantic concepts are composed of simpler semantic concepts and ultimately connected to NL words and phrases. Börschinger et al.'s approach instead produces NL groundings at the level of atomic MR constituents, which causes an explosion in the number of PCFG productions for complex MR languages. We estimated that Börschinger et al.'s approach would require more than 20! ($> 10^{18}$) productions for our navigation problem.[1] On the other hand, our method, which uses correspondences from the LHG at the semantic concept level, constructs a more focused PCFG of tractable size. It then extracts the MR for a novel sentence from the most-probable parse tree for the resulting PCFG. Our approach can produce a large, combinatorial number of different MRs for a wide range of novel sentences by composing relevant MR components from the resulting parse tree, whereas Börschinger et al.'s approach is only able to output MRs that are explicitly included as a nonterminals in the original learned PCFG.

The remainder of the paper is organized as follows. Section 2 reviews Börschinger et al.'s PCFG approach as well as the navigation task and data. Section 3 describes our enhanced PCFG approach and Section 4 presents an experimental evaluation of it. Then, Section 5 discusses the unique aspects of our approach and Section 6 describes additional related work. Finally, Section 7 presents future research directions and Section 8 gives our conclusions.

2 Background

2.1 Existing PCFG Approach

Our approach extends that of Börschinger et al. (2011), which in turn was inspired by a series of previous techniques (Lu et al., 2008; Liang et al., 2009; Kim and Mooney, 2010) following the idea of constructing correspondences between NL and MR in a single probabilistic generative framework. Particularly, their approach automatically constructs a PCFG that generates NL sentences from MRs, which indicates how atomic MR constituents are probabilistically related to NL words. The nonterminals in the grammar correspond to complete MRs, MR constituents, and NL phrases. The nonterminal for a composite MR generates each of its MR constituents, and each atomic MR, x, generates an NL phrase, $Phrase_x$. Each $Phrase_x$ then generates a sequence of $Word_x$'s for describing x, and each $Word_x$ can generate each possible word in the natural language. This allows the system to learn the words and phrases used to describe each atomic MR by properly weighting these rules. Figure 1 shows one possible derivation tree for a sample NL-MR pair and the PCFG rules that are constructed for it. Once a set of productions are assembled, their probabilities are learned using the Inside-Outside algorithm. Computing the most probable parse for a novel sentence with the trained PCFG provides its

[1] The corpus contains quite a few examples with landmarks plans containing more than 20 actions. This results in at least 20! permutations representing possible alignments between actions and NL words.

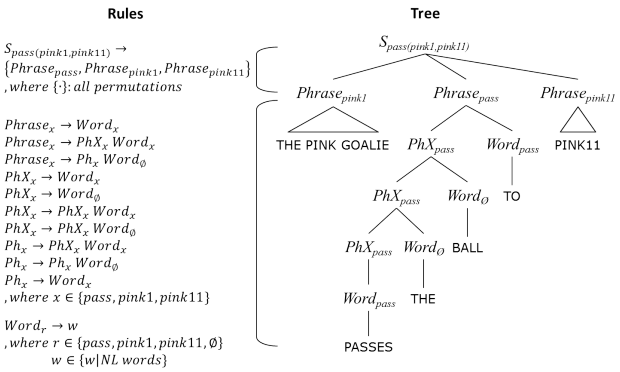


Figure 1: Derivation tree for the NL/MR pair: THE PINK GOALIE PASSES THE BALL TO PINK11 / $pass(pink1, pink11)$. Left side shows PCFG rules that are added for each stage (full MR to atomic MRs, and atomic MRs to NL words).

preferred MR interpretation in the topmost nonterminal.

Unfortunately, as discussed earlier, this approach only works for finite MR languages, and the grammar becomes intractably large even for finite but complex MRs. It effectively assumes that MRs are fairly small and includes every possible MR constituent as a nonterminal in the PCFG. This is not tractable for more complex MRs. Therefore, our extension incorporates a learned lexicon to constrain the space of productions, thereby making the size of the PCFG tractable for complex MRs, and even giving it the ability to handle infinite MR languages. Moreover, when processing novel sentences, our approach can produce a large space of novel MRs that were not anticipated during training, which is not the case for Börschinger et al.'s approach.

2.2 Navigation Task and Dataset

We employ the task and data introduced by Chen and Mooney (2011) whose goal is to interpret and follow NL navigation instructions in a virtual world. Figure 2 shows a sample execution path in a particular virtual world. The challenge is learning to perform this task by simply observing humans following instructions. Formally, given training data of the form $\{(e_1, a_1, w_1), \ldots, (e_n, a_n, w_n)\}$, where e_i is an 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).

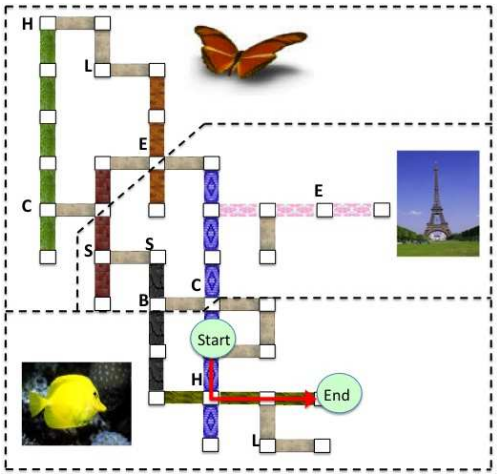


Figure 2: Sample virtual world from Chen and Mooney (2011) of interconnecting hallways with different floor and wall patterns and objects indicated by letters (e.g. "H" for hatrack).

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 3: Sample instruction with its constructed landmarks plan, components in bold compose the correct plan.

In order to learn, their system infers the intended formal plan p_i (the MR for a sentence) which produced the action sequence a_i from the instruction e_i. However, there is a large space of possible plans for any given action sequence. Chen and Mooney first construct a formal *landmarks plan*, c_i, for each a_i, which is a graph representing the context of every action and the world-state encountered during the execution of the sequence. The correct plan MR, p_i, is assumed to be a subgraph of c_i, and this causes a combinatorial matching problem between e_i and c_i in order to learn the correct meaning of e_i among all the possible subgraphs of c_i. The landmarks and correct plans for a sample instruction are shown in Figure 3, illustrating the complexity of the MRs.

Instead of directly solving the combinatorial correspondence problem, they first learn a semantic lex-

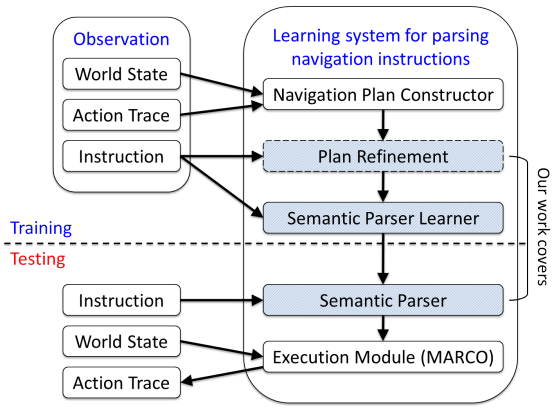


Figure 4: An overview of Chen and Mooney (2011)'s system. Our method replaces the plan refinement and semantic parser parts.

icon that maps words and short phrases to small subgraphs representing their inferred meanings from the (e_i, c_i) pairs. The lexicon is learned by evaluating pairs of n-grams, w_j, and MR graphs, m_j, and scoring them based on how much more likely m_j is a subgraph of the context c_i when w occurs in the corresponding instruction e_i. This process is similar to other "cross-situational" approaches to learning word meanings (Siskind, 1996; Thompson and Mooney, 2003). Then, a *plan refinement* step estimates p_i from c_i by greedily selecting high-scoring lexemes of the form (w_j, m_j) whose words and phrases (w_j) cover the instruction e_i and introduce components (m_j) from the landmarks plan c_i. The refined plans are used to construct supervised training data (e_i, p_i) for a supervised semantic-parser learner. The trained semantic parser can parse a novel instruction into a formal plan, which is finally executed for end-to-end evaluation. Figure 4 illustrates the overall system.

As this figure indicates, our new PCFG method replaces the plan refinement and semantic parser components in their system with a unified model that both 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.[2]

[2]In our experiments, we used the top 1,000 lexemes learned by Chen and Mooney (2011).

3 Our PCFG Approach

Like Börschinger et al. (2011), our approach learns a semantic parser directly from ambiguous supervision, specifically NL instructions paired with their complete landmarks plans as context. Our method incorporates the semantic lexemes as building blocks to find correspondences between NL words and semantic concepts represented by the lexeme MRs, instead of building connections between NL words and every possible MR constituent as in Börschinger et al.'s approach. Particularly, we utilize the hierarchical subgraph relationships between the MRs in the learned semantic lexicon to produce a smaller, more focused set of PCFG rules.[3] The intuition behind our approach is analogous to the hierarchical relations between nonterminals in syntactic parsing, where higher-level categories such as S, VP, or NP are further divided into smaller categories such as V, N, or Det, thereby forming a hierarchical structure. Inspired by this idea, we introduce a directed acyclic graph called the Lexeme Hierarchy Graph (LHG) which represents the hierarchical relationships between lexeme MRs. Since complex lexeme MRs represent complicated semantic concepts while simple MRs represent simple concepts, it is natural to construct a hierarchy amongst them. The LHGs for all of the training examples are used to construct production rules for the PCFG, which are then parametrized using EM. Finally, a novel sentence is semantically parsed by computing its most-probable parse using the trained PCFG, and then its MR is extracted from the resulting parse tree.

3.1 Constructing a Lexeme Hierarchy Graph

An LHG represents the hierarchy of lexical meanings relevant to a particular training instance by encoding the subgraph relations between the MRs of relevant lexemes. Algorithm 1 describes how an LHG is constructed for an ambiguous training pair of a sentence and its corresponding context, (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

[3]The total number of PCFG rules constructed for our navigation training sets is about 18,000, while Börschinger et al.'s method produces 33,000 rules for the much 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)

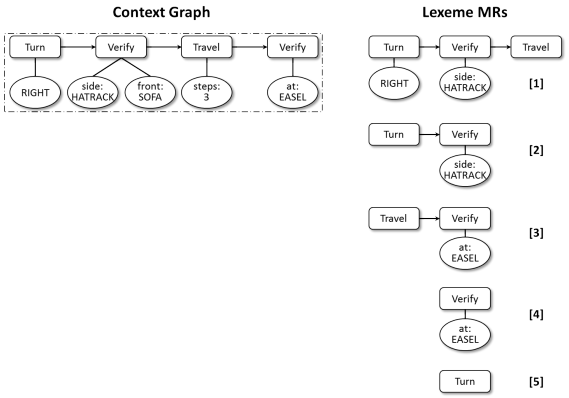
Find relevant lexemes $(w_1^i, m_1^i), \ldots, (w_n^i, m_n^i)$
  s.t. $m_j^i \subset c_i$
Create a starting node T; $MR(T) \leftarrow c_i$
for all m_j^i in the descending order of size **do**
  Create a node T_j^i; $MR(T_j^i) \leftarrow m_j^i$
  PLACELEXEME(T_j^i,T)
end for

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

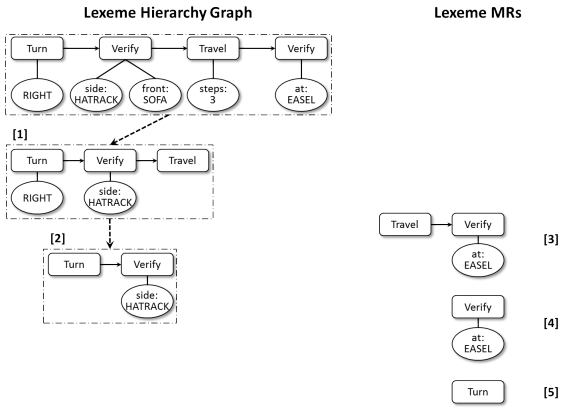
sorted in descending order based on the number of nodes in their MRs m_j^i. Then, after setting the context c_i as the MR of the root node ($MR(T) \leftarrow c_i$), lexemes are inserted, in order, into the graph to create a hierarchy of MRs, where each child's MR is a subgraph of the MR of each of its parents. Figure 5 illustrates a sample construction of an LHG for the following landmarks plan (c_i):

```
Turn(RIGHT),
Verify(side:HATRACK, front:SOFA),
Travel(steps:3),
Verify(at:EASEL)
```

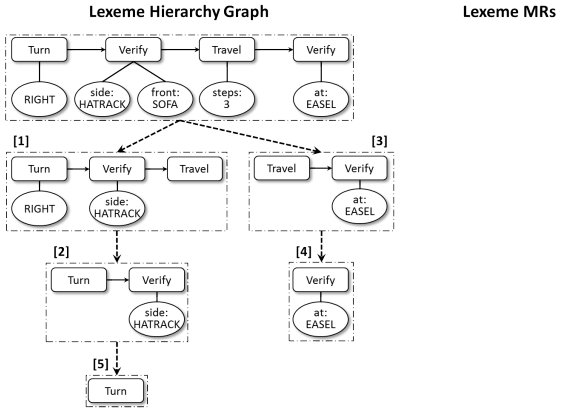
The initial LHG may contain nodes with too many children. This is a problem, because when we subsequently extract PCFG rules, we need to add a production for every k-permutation of the children of each node (see Section 3.2). To reduce the branching factor in the LHG, we introduce *pseudo-lexeme* nodes by repeatedly combining the two most similar children of each node. Pseudocode for the process is shown in Algorithm 2. The MR for a pseudo-lexeme is the minimal graph, m', that is a supergraph of both of the lexeme MRs that it combines. The pair of



(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], and MR [5] is recursively filtered down and added under [2].

Figure 5: Sample LHG construction.

Algorithm 2 ADDING PSEUDO LEXEMES TO LHG

Input: LHG with root T
Output: LHG with pseudo lexemes added
procedure RECONSTRUCTLHG(T)
  repeat
    $((T_i, T_j), m') \leftarrow$ pick the most similar pair (T_i, T_j) of children of T and the minimal extension m' s.t. $MR(T_i) \subset m'$, $MR(T_j) \subset m'$, $m' \subset MR(T)$
    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

most similar children, (m_i, m_j), is determined by measuring the fraction of the nodes in m_i and m_j that overlap with their minimum extension m' and is calculated as follows:

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

where $|m|$ is the number of nodes in the MR m. Adding pseudo-lexemes also has another advantage. They can be considered to be higher-level semantic concepts composed of two or more subconcepts. These higher-level concepts will likely occur in other training examples as well, which allows for more flexible interpretations. For example, assuming the rule $A \rightarrow BCD$ is constructed from an LHG, we will introduce a pseudo lexeme E and build two rules $A \rightarrow BE$ and $E \rightarrow CD$. It is likely that E also occurs in another rule constructed from other training examples such as $E \rightarrow FG$. This increases the model's expressive power by supporting additional derivations such as $A \rightarrow^* BFG$, providing more flexibility when parsing novel NL sentences.

3.2 Composing PCFG Rules

The next step composes PCFG rules from the LHGs and is summarized in Figure 6. 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,

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

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

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

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

Figure 6: Summary of the rule generation process. *NLs* refer to the set of NL words in the corpus. Lexeme rules come from the schemata of Börschinger et al. (2011), and allow every lexeme MR to generate one or more NL words. Note that pseudo-lexeme nodes do not produce NL words.

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 $S_m \rightarrow S_{m_i}...S_{m_j}$, where the RHS is some k-permutation of the nonterminals for the MRs of the children of node T. Börschinger et al. assume that every atomic MR generates at least one NL word. However, since we do not know which subgraph of the overall context (i.e. c_i, the MR of the root node) conveys the intended plan and is therefore expressed in the NL instruction, we must allow each ordered subset of the children of a node (i.e. each k-permutation) to be a possible generation.

The rest of the process more closely follows Börschinger et al.'s. Every MR, m, of a lexeme node[4] 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. have $Word_m$ generate all possible NL words (each of which are

[4]We exclude pseudo-lexeme nodes in this process, because they should only generate words through generating lexemes.

subsequently weighted by EM training), in our approach, each $Word_m$ only produces the NL phrase associated with m in the lexicon, or individual words that appear in this phrase. The words not covered by $Word_m$ also can be generated by $Word_\emptyset$ which has rules for every word. Ph_m and PhX_m ensure that $Phrase_m$ produces at least one $Word_m$, where PhX_m indicates that one or more $Word_m$'s have already been generated, and Ph_m indicates that no $Word_m$ has yet been generated.

3.3 Parsing Novel NL Sentences

To learn the parameters of the resulting PCFG, we use the Inside-Outside algorithm.[5] Then, the standard probabilistic CKY algorithm is used to produce the most probable parse for novel NL sentences (Jurafsky and 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. However, in our approach, the correct MR is constructed by properly composing the appropriate subset of lexeme MRs from the most-probable parse tree. This allows the system to produce a wide variety of novel MRs for novel sentences, as long as the correct MR is a subgraph of the complete context (c_i) for at least one of the training sentences.

First, the parse tree is pruned to remove all subtrees starting with $Phrase_x$ nodes. This leaves a tree consisting of the $Root$ and a set of S_m nodes. The pruned subtrees only concern generating NL words and phrases from the selected MRs. The remaining tree shows which MR constituents were selected from the available context, from which the sentence is then generated. Each leaf in the pruned tree represents an MR constituent that was used to generate a phrase in the sentence. These are the constituents we want to assemble and compose into a final MR for the sentence.

Algorithm 3 describes the procedure for extracting the final MR from the pruned parse tree. Figure 7 graphically depicts a sample trace of this algorithm. 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 iff its corresponding node in any of the children's MRs were marked. The final output is the MR constructed by removing all of the unmarked nodes from the MR for the root node.

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 Experimental Evaluation

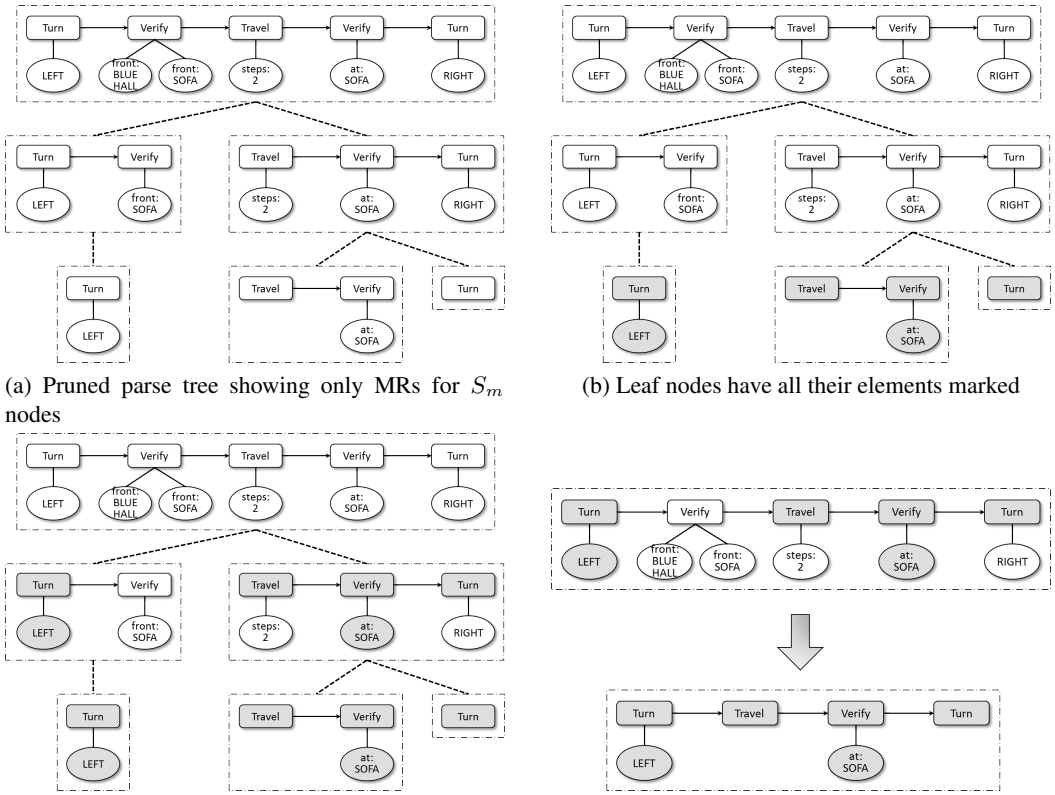
For evaluation, we used the same data and methodology as Chen and Mooney (2011). Please see their paper for more details.

4.1 Data

We used the English instructions and follower data collected by MacMahon et al. (2006).[6] 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 spread evenly across the three worlds, and there were 1 to 15 human followers for each instruction who executed an average of 10.4 actions per instruction. Each instruction is a paragraph consisting of an average of 5.0 sentences, each containing an average of 7.8 words. Chen and Mooney constructed the additional single-sentence corpus by matching each sentence with the majority of human

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

[6] Available at `http://www.cs.utexas.edu/users/ml/clamp/navigation/`



(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 7: Sample construction of MR output from pruned parse tree.

followers' actions. We use this single-sentence version for training, but use both the single-sentence and the original paragraph version for testing. Each sentence was manually annotated with a "gold standard" execution plan, which is used for evaluation but not for training.

4.2 Methodology and Results

Experiments were conducted using "leave one environment out" cross-validation, training on two environments and testing on the third, averaging over all three test environments. We perform direct comparison to the best results of Chen and Mooney (2011) (referred to as CM). A Wilcoxon signed-rank test is performed for statistical significance, and '$*$' denotes significant differences ($p < .01$) in the tables.

Semantic Parsing Results

We first evaluated how well our system learns to map novel NL sentences for new test environments into their correct MRs. Partial semantic-parsing accuracy (Chen and Mooney, 2011) is calculated by comparing the system's MR output to the hand-annotated gold standard. Accuracy is measured in terms of precision, recall, and F1 for individual MR constituents (thereby awarding partial credit for approximately correct MRs).

	Precision	Recall	F1
Our system	87.58	$*$65.41	$*$**74.81**
CM	$*$90.22	55.10	68.37

Table 1: Test accuracy for semantic parsing. '$*$' denotes difference is statistically significant.

Table 1 demonstrates that our method outperforms CM by 6 points in F1. Our PCFG-based approach 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 CM, since they lose potentially useful information, particularly during the refinement stage, due to the separate disjoint components of the system.

	Single-sentence	Paragraph
Our system	∗**57.22%**	∗**20.17%**
CM	54.40%	16.18%

Table 2: Successful plan execution rates for novel test data. '∗' means statistical significance.

Navigation Plan Execution Results

Next, we test the end-to-end system by executing the parsed navigation plans for test instructions in novel environments to see if they reach the exact desired destinations in the environment. Table 2 shows the successful end-to-end navigation-task completion rate for both single-sentences and complete paragraph instructions.

Again, our system outperforms CM'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 the redundancy in the 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.

5 Discussion

Our approach improves on Börschinger et al. (2011)'s method in the following ways:

• The building blocks for associating NL and MR are semantic lexemes instead of atomic MR constituents. This prevents the number of constructed PCFG rules from becoming intractably large as happens with Börschinger et al.'s approach. As previously mentioned, lexeme MRs are intuitively analogous to syntactic categories in that complex lexeme MRs represent complicated semantic concepts whereas higher-level syntactic categories such as S, VP, or NP represent complex syntactic structures.

• Our approach has the ability to produce previously unseen MRs, whereas Börschinger et al. can only generate an MR if it is explicitly included in the PCFG rules constructed from the training data. Even though our MR parse is restricted to be a subgraph of some training context, c_i, our model allows for exponentially many combinations.

In addition, our approach can produce a wider range of MR outputs than Chen and Mooney (2011)'s even though we use their semantic lexicon as input. Their system deterministically builds a supervised training set by greedily selecting high-scoring lexemes, thus implicitly including only high-scoring lexemes during training. On the other hand, our probabilistic approach also considers relatively low-scoring but useful lexemes, thereby utilizing more semantic concepts in the lexicon. In particular, this explains why our approach obtains higher *recall* in the evaluation of semantic parsing.

Even though we have demonstrated our approach on the specific task of following navigation instructions, it is straightforward to apply it to other language-grounding tasks where NL sentences potentially refer to some subset of states, events, or actions in the world, as long as this overall context can be represented as a semantic graph or logical form. Since the semantic lexicon is an input to our system, other approaches to lexicon learning are also easily incorporated.

6 Related Work

Most work on learning semantic parsers that map natural-language sentences to formal representations of their meaning have relied upon totally supervised training data consisting of NL/MR pairs (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Kate and Mooney, 2006; Wong and Mooney, 2007; Zettlemoyer and Collins, 2007; Lu et al., 2008; Zettlemoyer and Collins, 2009). Several recent approaches have investigated grounded learning from ambiguous supervision extracted from perceptual context. A number of approaches (Kate and Mooney, 2007; Chen and Mooney, 2008; Chen et al., 2010; Kim and Mooney, 2010; Börschinger et al., 2011) assume training data consisting of a set of sentences each associated with a small set of MRs, one of which is usually the correct meaning of the sentence. Many of these approaches (Kate and Mooney, 2007; Chen and Mooney, 2008; Chen et al., 2010) disambiguate the data and match NL sentences to their correct MR by iteratively retraining a supervised semantic parser. Kim and Mooney (2010) proposed a generative semantic parsing model that first chooses which MRs to describe and then generates a hybrid tree structure (Lu et al., 2008) containing both the MR and NL sentence. They train

this model on ambiguous data using EM. As previously discussed, Börschinger et al. (2011) use a PCFG generative model and also train it on ambiguous data using EM. Liang et al. (2009) assume each sentence maps to one *or more* semantic *records* (i.e. MRs) and trains a hierarchical semi-Markov generative model using EM, and then finds a Viterbi alignment between NL words and records and their constituents. Several recent projects (Branavan et al., 2009; Vogel and Jurafsky, 2010) use NL instructions to guide *reinforcement learning* from independent exploration with delayed rewards. These systems do not even need the ambiguous supervision obtained from observing humans follow instructions; however, they do not learn semantic parsers that map sentences to complex, structural representations of their meaning.

Interpreting and executing NL navigation instructions is our primary task, and several other recent projects have studied related problems. Shimizu and Haas (2009) present a system that parses natural language instructions into actions. However, they limit the number of possible actions to only 15 and treat the problem as a sequence labeling problem that is solved using a CRF with supervised training. Matuszek et al. (2010) developed a system that learns to map NL instructions to executable commands for a robot navigating in an environment constructed by a laser range finder. However, their approach has limitations of ignoring any objects or other landmarks in the environment to which the instructions can refer. There are several recent projects (Vogel and Jurafsky, 2010; Kollar et al., 2010; Tellex et al., 2011) which learn to follow instructions in more linguistically complex environments. However, they assume predefined spatial words, direct matching between NL words and the names of objects and other landmarks in the MR, and/or an existing syntactic parser. By contrast, our work does not assume any prior linguistic knowledge, syntactic, lexical, or semantic, and must *learn* the mapping between NL words and phrases and the MR terms describing landmarks.

7 Future Work

In the future, we would like to develop a better lexicon learner since our PCFG approach critically relies on the quality of the learned lexicon. Particularly, we would like to investigate how syntactic information (such as part-of-speech tags induced using unsupervised learning) could be used to improve semantic-lexicon learning. For example, some of the current lexicon entries violate the general constraint that nouns usually refer to objects and verbs to actions. Ideally, the lexicon learner would be able to induce and then utilize this sort of relationship between syntax and semantics.

In addition, we want to investigate the use of *discriminative reranking* (Collins, 2000), which has proven effective in various other NLP tasks. We would expect the final MR output to improve if a discriminative model, which uses additional global features, is used to rerank the top-k parses produced by our generative PCFG model.

8 Conclusions

We have presented a novel method for learning a semantic parser given only highly ambiguous supervision. Our model enhances Börschinger et al. (2011)'s approach to reducing the problem of grounded learning of semantic parsers to PCFG induction. We use a learned semantic lexicon to aid the construction of a smaller and more focused set of PCFG productions. This allows the approach to scale to complex MR languages that define a large (potentially infinite) space of representations for capturing the meaning of sentences. By contrast, the previous PCFG approach requires a finite MR language and its grammar grows intractably large for even moderately complex MR languages. In addition, our algorithm for composing MRs from the final parse tree provides the flexibility to produce a wide range of novel MRs that were not seen during training. Evaluations on a previous corpus of navigational instructions for virtual environments has demonstrated the effectiveness of our method compared to a recent competing system.

Acknowledgments

We thank the anonymous reviewers and David Chen for useful comments that helped improve this paper. This work was funded by NSF grants IIS-0712907 and IIS-1016312. Experiments were performed on the Mastodon Cluster, provided by NSF grant EIA-0303609.

References

Benjamin Börschinger, Bevan K. Jones, and Mark Johnson. 2011. Reducing grounded learning tasks to grammatical inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-11)*, pages 1416–1425, Stroudsburg, PA, USA. Association for Computational Linguistics.

S.R.K. Branavan, Harr Chen, Luke S. Zettlemoyer, and Regina Barzilay. 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.

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proceedings of 25th International Conference on Machine Learning (ICML-2008)*, Helsinki, Finland, July.

David L. Chen and Raymond J. Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-11)*, San Francisco, CA, USA, August.

David L. Chen, Joohyun Kim, and Raymond J. Mooney. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37:397–435.

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

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, NJ.

R. J. Kate and R. J. Mooney. 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)*, pages 913–920, Sydney, Australia, July.

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

Joohyun Kim and Raymond. J. Mooney. 2010. Generative alignment and semantic parsing for learning from ambiguous supervision. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-10)*, pages 543–551. Association for Computational Linguistics.

Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. 2010. Toward understanding natural language directions. In *Proceedings of Human Robot Interaction Conference (HRI-2010)*.

P. Liang, M. I. Jordan, and D. Klein. 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.

Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*, pages 783–792, Morristown, NJ, USA. Association for Computational Linguistics.

M. MacMahon, B. Stankiewicz, and B. Kuipers. 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, July.

Cynthia Matuszek, Dieter Fox, and Karl Koscher. 2010. Following directions using statistical machine translation. In *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction (HRI-10)*, pages 251–258, New York, NY, USA. ACM.

Nobuyuki Shimizu and Andrew Haas. 2009. Learning to follow navigational route instructions. In *Proceedings of the Twenty First International Joint Conference on Artificial Intelligence (IJCAI-2009)*.

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

Stefanie Tellex, Thomas Kolla, Steven Dickerson, Matthew R. Walter, Ashis G. Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-11)*, August.

Cynthia A. Thompson and Raymond J. Mooney. 2003. Acquiring word-meaning mappings for natural language interfaces. *Journal of Artificial Intelligence Research*, 18:1–44.

Adam Vogel and Dan Jurafsky. 2010. Learning to follow navigational directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*.

Yuk Wah Wong and Raymond J. Mooney. 2007. Learning synchronous grammars for semantic parsing with

lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, pages 960–967, Prague, Czech Republic, June.

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

Luke S. Zettlemoyer and Michael Collins. 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, July.

Luke S. Zettlemoyer and Michael Collins. 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)*, pages 678–687, Prague, Czech Republic, June.

Luke .S. Zettlemoyer and Micheal Collins. 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 (ACL-IJCNLP-09)*, pages 976–984. Association for Computational Linguistics.