In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL-2006)*. pp. 439-446, New York City, NY, June 2006.

Learning for Semantic Parsing with Statistical Machine Translation

Yuk Wah Wong and Raymond J. Mooney
Department of Computer Sciences
The University of Texas at Austin
1 University Station C0500
Austin, TX 78712-0233, USA
{ywwong,mooney}@cs.utexas.edu

Abstract

We present a novel statistical approach to semantic parsing, WASP, for constructing a complete, formal meaning representation of a sentence. A semantic parser is learned given a set of sentences annotated with their correct meaning representations. The main innovation of WASP is its use of state-of-the-art statistical machine translation techniques. A word alignment model is used for lexical acquisition, and the parsing model itself can be seen as a syntax-based translation model. We show that WASP performs favorably in terms of both accuracy and coverage compared to existing learning methods requiring similar amount of supervision, and shows better robustness to variations in task complexity and word order.

1 Introduction

Recent work on natural language understanding has mainly focused on shallow semantic analysis, such as semantic role labeling and word-sense disambiguation. This paper considers a more ambitious task of *semantic parsing*, which is the construction of a complete, formal, symbolic, *meaning representation* (MR) of a sentence. Semantic parsing has found its way in practical applications such as natural-language (NL) interfaces to databases (Androutsopoulos et al., 1995) and advice taking (Kuhlmann et al., 2004). Figure 1 shows a sample MR written in a *meaning-representation language* (MRL) called CLANG, which is used for encoding coach advice given to simulated soccer-playing agents (Kuhlmann et al., 2004).

```
((bowner our {4})
 (do our {6} (pos (left (half our)))))
```

If our player 4 has the ball, then our player 6 should stay in the left side of our half.

Figure 1: A meaning representation in CLANG

Prior research in semantic parsing has mainly focused on relatively simple domains such as ATIS (Air Travel Information Service) (Miller et al., 1996; Papineni et al., 1997; Macherey et al., 2001), in which a typical MR is only a single semantic frame. Learning methods have been devised that can generate MRs with a complex, nested structure (cf. Figure 1). However, these methods are mostly based on deterministic parsing (Zelle and Mooney, 1996; Kate et al., 2005), which lack the robustness that characterizes recent advances in statistical NLP. Other learning methods involve the use of fully-annotated augmented parse trees (Ge and Mooney, 2005) or prior knowledge of the NL syntax (Zettlemoyer and Collins, 2005) in training, and hence require extensive human efforts when porting to a new domain or language.

In this paper, we present a novel statistical approach to semantic parsing which can handle MRs with a nested structure, based on previous work on semantic parsing using transformation rules (Kate et al., 2005). The algorithm learns a semantic parser given a set of NL sentences annotated with their correct MRs. It requires no prior knowledge of the NL syntax, although it assumes that an unambiguous, context-free grammar (CFG) of the target MRL is available. The main innovation of this al-

```
answer(count(city(loc_2(countryid(usa)))))
```
How many cities are there in the US?

Figure 2: A meaning representation in GEOQUERY

gorithm is its integration with state-of-the-art statistical machine translation techniques. More specifically, a statistical word alignment model (Brown et al., 1993) is used to acquire a bilingual lexicon consisting of NL substrings coupled with their translations in the target MRL. Complete MRs are then formed by combining these NL substrings and their translations under a parsing framework called the synchronous CFG (Aho and Ullman, 1972), which forms the basis of most existing statistical syntax-based translation models (Yamada and Knight, 2001; Chiang, 2005). Our algorithm is called WASP, short for *Word Alignment-based Semantic Parsing*. In initial evaluation on several real-world data sets, we show that WASP performs favorably in terms of both accuracy and coverage compared to existing learning methods requiring the same amount of supervision, and shows better robustness to variations in task complexity and word order.

Section 2 provides a brief overview of the domains being considered. In Section 3, we present the semantic parsing model of WASP. Section 4 outlines the algorithm for acquiring a bilingual lexicon through the use of word alignments. Section 5 describes a probabilistic model for semantic parsing. Finally, we report on experiments that show the robustness of WASP in Section 6, followed by the conclusion in Section 7.

2 Application Domains

In this paper, we consider two domains. The first domain is ROBOCUP. ROBOCUP (`www.robocup.org`) is an AI research initiative using robotic soccer as its primary domain. In the ROBOCUP Coach Competition, teams of agents compete on a simulated soccer field and receive coach advice written in a formal language called CLANG (Chen et al., 2003). Figure 1 shows a sample MR in CLANG.

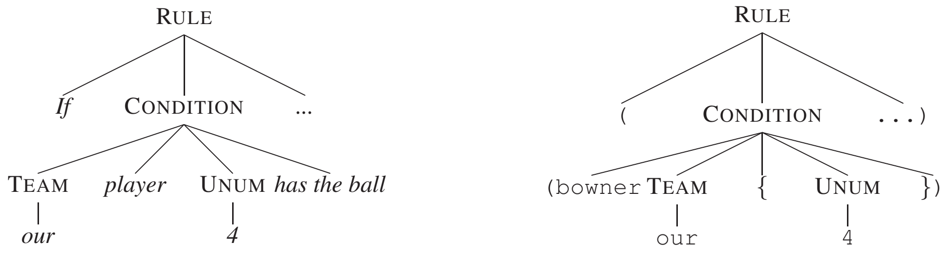
The second domain is GEOQUERY, where a functional, variable-free query language is used for querying a small database on U.S. geography (Zelle and Mooney, 1996; Kate et al., 2005). Figure 2 shows a sample query in this language. Note that both domains involve the use of MRs with a complex, nested structure.

3 The Semantic Parsing Model

To describe the semantic parsing model of WASP, it is best to start with an example. Consider the task of translating the sentence in Figure 1 into its MR in CLANG. To achieve this task, we may first analyze the syntactic structure of the sentence using a *semantic grammar* (Allen, 1995), whose non-terminals are the ones in the CLANG grammar. The meaning of the sentence is then obtained by combining the meanings of its sub-parts according to the semantic parse. Figure 3(a) shows a possible partial semantic parse of the sample sentence based on CLANG non-terminals (UNUM stands for uniform number). Figure 3(b) shows the corresponding CLANG parse from which the MR is constructed.

This process can be formalized as an instance of *synchronous parsing* (Aho and Ullman, 1972), originally developed as a theory of compilers in which syntax analysis and code generation are combined into a single phase. Synchronous parsing has seen a surge of interest recently in the machine translation community as a way of formalizing syntax-based translation models (Melamed, 2004; Chiang, 2005). According to this theory, a semantic parser defines a *translation*, a set of pairs of strings in which each pair is an NL sentence coupled with its MR. To finitely specify a potentially infinite translation, we use a *synchronous context-free grammar* (SCFG) for generating the pairs in a translation. Analogous to an ordinary CFG, each SCFG rule consists of a single non-terminal on the left-hand side (LHS). The right-hand side (RHS) of an SCFG rule is a pair of strings, $\langle \alpha, \beta \rangle$, where the non-terminals in β are a permutation of the non-terminals in α. Below are some SCFG rules that can be used for generating the parse trees in Figure 3:

$$\begin{aligned}
\textsc{Rule} &\rightarrow \langle \textit{if}\ \textsc{Condition}_{\boxed{1}}\ ,\ \textsc{Directive}_{\boxed{2}}\ .\ ,\\
&\qquad \texttt{(}\textsc{Condition}_{\boxed{1}}\ \textsc{Directive}_{\boxed{2}}\texttt{)} \rangle \\
\textsc{Condition} &\rightarrow \langle \textsc{Team}_{\boxed{1}}\ \textit{player}\ \textsc{Unum}_{\boxed{2}}\ \textit{has the ball}\ ,\\
&\qquad \texttt{(bowner}\ \textsc{Team}_{\boxed{1}}\ \{\textsc{Unum}_{\boxed{2}}\}\texttt{)} \rangle \\
\textsc{Team} &\rightarrow \langle \textit{our}\ ,\ \texttt{our} \rangle \\
\textsc{Unum} &\rightarrow \langle \textit{4}\ ,\ \texttt{4} \rangle
\end{aligned}$$



(a) English

(b) CLANG

Figure 3: Partial parse trees for the CLANG statement and its English gloss shown in Figure 1

Each SCFG rule $X \rightarrow \langle \alpha, \beta \rangle$ is a combination of a production of the NL semantic grammar, $X \rightarrow \alpha$, and a production of the MRL grammar, $X \rightarrow \beta$. Each rule corresponds to a *transformation rule* in Kate et al. (2005). Following their terminology, we call the string α a *pattern*, and the string β a *template*. Non-terminals are indexed to show their association between a pattern and a template. All derivations start with a pair of associated start symbols, $\langle S_{\boxed{1}}, S_{\boxed{1}} \rangle$. Each step of a derivation involves the rewriting of a pair of associated non-terminals in both of the NL and MRL streams. Below is a derivation that would generate the sample sentence and its MR simultaneously: (Note that RULE is the start symbol for CLANG)

$\langle \textsc{Rule}_{\boxed{1}}, \textsc{Rule}_{\boxed{1}} \rangle$
$\Rightarrow \langle$ *if* $\textsc{Condition}_{\boxed{1}}$, $\textsc{Directive}_{\boxed{2}}$. ,
(`(` $\textsc{Condition}_{\boxed{1}}$ $\textsc{Directive}_{\boxed{2}}$`)` $\rangle$
$\Rightarrow \langle$ *if* $\textsc{Team}_{\boxed{1}}$ *player* $\textsc{Unum}_{\boxed{2}}$ *has the ball,* $\textsc{Dir}_{\boxed{3}}$. ,
`((bowner` $\textsc{Team}_{\boxed{1}}$ $\{\textsc{Unum}_{\boxed{2}}\}$`)` $\textsc{Dir}_{\boxed{3}}$`)` $\rangle$
$\Rightarrow \langle$ *if our player* $\textsc{Unum}_{\boxed{1}}$ *has the ball,* $\textsc{Dir}_{\boxed{2}}$. ,
`((bowner our` $\{\textsc{Unum}_{\boxed{1}}\}$`)` $\textsc{Dir}_{\boxed{2}}$`)` $\rangle$
$\Rightarrow \langle$ *if our player 4 has the ball,* $\textsc{Directive}_{\boxed{1}}$. ,
`((bowner our {4})` $\textsc{Directive}_{\boxed{1}}$`)` $\rangle$
$\Rightarrow$ ...
$\Rightarrow \langle$ *if our player 4 has the ball, then our player 6 should stay in the left side of our half.* ,
`((bowner our {4})`
`(do our {6} (pos (left (half our)))))` $\rangle$

Here the MR string is said to be a *translation of* the NL string. Given an input sentence, $\mathbf{e}$, the task of semantic parsing is to find a derivation that yields $\langle \mathbf{e}, \mathbf{f} \rangle$, so that $\mathbf{f}$ is a translation of $\mathbf{e}$. Since there may be multiple derivations that yield $\mathbf{e}$ (and thus multiple possible translations of $\mathbf{e}$), a mechanism must be devised for discriminating the correct derivation from the incorrect ones.

The semantic parsing model of WASP thus consists of an SCFG, G, and a probabilistic model, parameterized by λ, that takes a possible derivation, $\mathbf{d}$, and returns its likelihood of being correct given an input sentence, $\mathbf{e}$. The output translation, $\mathbf{f}^{\star}$, for a sentence, $\mathbf{e}$, is defined as:

$$\mathbf{f}^{\star} = m \left(\arg\max_{\mathbf{d} \in D(G|\mathbf{e})} \Pr_{\lambda}(\mathbf{d}|\mathbf{e}) \right) \tag{1}$$

where $m(\mathbf{d})$ is the MR string that a derivation $\mathbf{d}$ yields, and $D(G|\mathbf{e})$ is the set of all possible derivations of G that yield $\mathbf{e}$. In other words, the output MR is the yield of the most probable derivation that yields $\mathbf{e}$ in the NL stream.

The learning task is to induce a set of SCFG rules, which we call a *lexicon*, and a probabilistic model for derivations. A lexicon defines the set of derivations that are possible, so the induction of a probabilistic model first requires a lexicon. Therefore, the learning task can be separated into two sub-tasks: (1) the induction of a lexicon, followed by (2) the induction of a probabilistic model. Both sub-tasks require a training set, $\{\langle \mathbf{e}_i, \mathbf{f}_i \rangle\}$, where each training example $\langle \mathbf{e}_i, \mathbf{f}_i \rangle$ is an NL sentence, $\mathbf{e}_i$, paired with its correct MR, $\mathbf{f}_i$. Lexical induction also requires an unambiguous CFG of the MRL. Since there is no lexicon to begin with, it is not possible to include correct derivations in the training data. This is unlike most recent work on syntactic parsing based on gold-standard treebanks. Therefore, the induction of a probabilistic model for derivations is done in an unsupervised manner.

4 Lexical Acquisition

In this section, we focus on lexical learning, which is done by finding optimal *word alignments* between

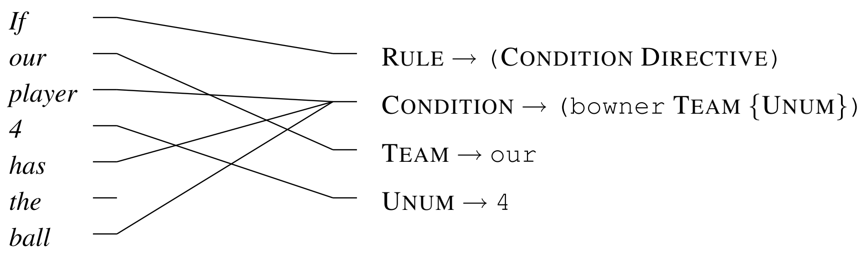

If
our
player
4
has
the
ball
Rule → (Condition Directive)
Condition → (bowner Team {Unum})
Team → our
Unum → 4

Figure 4: Partial word alignment for the CLang statement and its English gloss shown in Figure 1

NL sentences and their MRs in the training set. By defining a mapping of words from one language to another, word alignments define a bilingual lexicon. Using word alignments to induce a lexicon is not a new idea (Och and Ney, 2003). Indeed, attempts have been made to directly apply machine translation systems to the problem of semantic parsing (Papineni et al., 1997; Macherey et al., 2001). However, these systems make no use of the MRL grammar, thus allocating probability mass to MR translations that are not even syntactically well-formed. Here we present a lexical induction algorithm that guarantees syntactic well-formedness of MR translations by using the MRL grammar.

The basic idea is to train a statistical word alignment model on the training set, and then form a lexicon by extracting transformation rules from the $K = 10$ most probable word alignments between the training sentences and their MRs. While NL words could be directly aligned with MR tokens, this is a bad approach for two reasons. First, not all MR tokens carry specific meanings. For example, in CLang, parentheses and braces are delimiters that are semantically vacuous. Such tokens are not supposed to be aligned with any words, and inclusion of these tokens in the training data is likely to confuse the word alignment model. Second, MR tokens may exhibit polysemy. For instance, the CLang predicate `pt` has three meanings based on the types of arguments it is given: it specifies the xy-coordinates (e.g. `(pt 0 0)`), the current position of the ball (i.e. `(pt ball)`), or the current position of a player (e.g. `(pt our 4)`). Judging from the `pt` token alone, the word alignment model would not be able to identify its exact meaning.

A simple, principled way to avoid these difficulties is to represent an MR using a sequence of productions used to generate it. Specifically, the sequence corresponds to the top-down, left-most derivation of an MR. Figure 4 shows a partial word alignment between the sample sentence and the linearized parse of its MR. Here the second production, Condition → `(bowner` Team {Unum}`)`, is the one that rewrites the Condition non-terminal in the first production, Rule → `(`Condition Directive`)`, and so on. Note that the structure of a parse tree is preserved through linearization, and for each MR there is a unique linearized parse, since the MRL grammar is unambiguous. Such alignments can be obtained through the use of any off-the-shelf word alignment model. In this work, we use the GIZA++ implementation (Och and Ney, 2003) of IBM Model 5 (Brown et al., 1993).

Assuming that each NL word is linked to at most one MRL production, transformation rules are extracted in a bottom-up manner. The process starts with productions whose RHS is all terminals, e.g. Team → `our` and Unum → `4`. For each of these productions, $X \rightarrow \beta$, a rule $X \rightarrow \langle\alpha, \beta\rangle$ is extracted such that α consists of the words to which the production is linked, e.g. Team → ⟨*our*, `our`⟩, Unum → ⟨*4*, `4`⟩. Then we consider productions whose RHS contains non-terminals, i.e. predicates with arguments. In this case, an extracted pattern consists of the words to which the production is linked, as well as non-terminals showing where the arguments are realized. For example, for the `bowner` predicate, the extracted rule would be Condition → ⟨Team$_{\boxed{1}}$ *player* Unum$_{\boxed{2}}$ *has* (1) *ball*, `(bowner` Team$_{\boxed{1}}$ {Unum$_{\boxed{2}}$}`)`⟩, where (1) denotes a *word gap* of size 1, due to the unaligned word *the* that comes between *has* and *ball*. A word gap, (g), can be seen as a non-terminal that expands to at most g words in the NL stream, which allows for some flexibility in pattern matching. Rule extraction thus proceeds backward from the end of a linearized MR

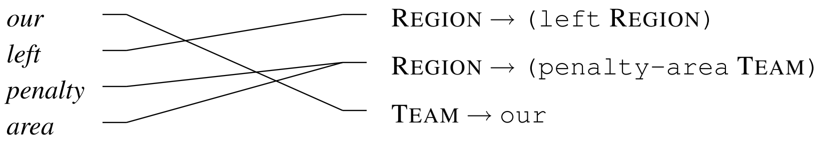


Figure 5: A word alignment from which no rules can be extracted for the `penalty-area` predicate

parse (so that a predicate is processed only after its arguments have all been processed), until rules are extracted for all productions.

There are two cases where the above algorithm would not extract any rules for a production r. First is when no descendants of r in the MR parse are linked to any words. Second is when there is a link from a word w, covered by the pattern for r, to a production r' outside the sub-parse rooted at r. Rule extraction is forbidden in this case because it would destroy the link between w and r'. The first case arises when a component of an MR is not realized, e.g. assumed in context. The second case arises when a predicate and its arguments are not realized close enough. Figure 5 shows an example of this, where no rules can be extracted for the `penalty-area` predicate. Both cases can be solved by merging nodes in the MR parse tree, combining several productions into one. For example, since no rules can be extracted for `penalty-area`, it is combined with its parent to form REGION → `(left (penalty-area` TEAM`))`, for which the pattern TEAM *left penalty area* is extracted.

The above algorithm is effective only when words linked to an MR predicate and its arguments stay close to each other, a property that we call *phrasal coherence*. Any links that destroy this property would lead to excessive node merging, a major cause of overfitting. Since building a model that strictly observes phrasal coherence often requires rules that model the reordering of tree nodes, our goal is to bootstrap the learning process by using a simpler, word-based alignment model that produces a generally coherent alignment, and then remove links that would cause excessive node merging before rule extraction takes place. Given an alignment, $\mathbf{a}$, we count the number of links that would prevent a rule from being extracted for each production in the MR parse. Then the total sum for all productions is obtained, denoted by $v(\mathbf{a})$. A greedy procedure is employed that repeatedly removes a link $a \in \mathbf{a}$ that would maximize $v(\mathbf{a}) - v(\mathbf{a}\backslash\{a\}) > 0$, until $v(\mathbf{a})$ cannot be further reduced. A link $w \leftrightarrow r$ is never removed if the translation probability, $\Pr(r|w)$, is greater than a certain threshold (0.9). To replenish the removed links, links from the most probable reverse alignment, $\tilde{\mathbf{a}}$ (obtained by treating the source language as target, and vice versa), are added to $\mathbf{a}$, as long as $\mathbf{a}$ remains n-to-1, and $v(\mathbf{a})$ is not increased.

5 Parameter Estimation

Once a lexicon is acquired, the next task is to learn a probabilistic model for the semantic parser. We propose a maximum-entropy model that defines a conditional probability distribution over derivations ($\mathbf{d}$) given the observed NL string ($\mathbf{e}$):

$$\Pr_\lambda(\mathbf{d}|\mathbf{e}) = \frac{1}{Z_\lambda(\mathbf{e})} \exp \sum_i \lambda_i f_i(\mathbf{d}) \qquad (2)$$

where f_i is a *feature function*, and $Z_\lambda(\mathbf{e})$ is a normalizing factor. For each rule r in the lexicon there is a feature function that returns the number of times r is used in a derivation. Also for each word w there is a feature function that returns the number of times w is generated from word gaps. Generation of unseen words is modeled using an extra feature whose value is the *total* number of words generated from word gaps. The number of features is quite modest (less than 3,000 in our experiments). A similar feature set is used by Zettlemoyer and Collins (2005).

Decoding of the model can be done in cubic time with respect to sentence length using the Viterbi algorithm. An Earley chart is used for keeping track of all derivations that are consistent with the input (Stolcke, 1995). The maximum conditional likelihood criterion is used for estimating the model parameters, λ_i. A Gaussian prior ($\sigma^2 = 1$) is used for regularizing the model (Chen and Rosenfeld, 1999). Since gold-standard derivations are not available in the training data, correct derivations must be treated as hidden variables. Here we use a version of improved iterative scaling (IIS) coupled with EM (Riezler et al., 2000) for finding an optimal set of parameters.[1] Unlike the fully-supervised case, the conditional likelihood is not concave with respect to λ, so the estimation algorithm is sensitive to initial parameters. To assume as little as possible, λ is initialized to $\mathbf{0}$. The estimation algorithm requires statistics that depend on all possible derivations for a sentence or a sentence-MR pair. While it is not feasible to enumerate all derivations, a variant of the Inside-Outside algorithm can be used for efficiently collecting the required statistics (Miyao and Tsujii, 2002). Following Zettlemoyer and Collins (2005), only rules that are used in the best parses for the training set are retained in the final lexicon. All other rules are discarded. This heuristic, commonly known as *Viterbi approximation*, is used to improve accuracy, assuming that rules used in the best parses are the most accurate.

6 Experiments

We evaluated Wasp in the RoboCup and Geoquery domains (see Section 2). To build a corpus for RoboCup, 300 pieces of coach advice were randomly selected from the log files of the 2003 RoboCup Coach Competition, which were manually translated into English (Kuhlmann et al., 2004). The average sentence length is 22.52. To build a corpus for Geoquery, 880 English questions were gathered from various sources, which were manually translated into the functional Geoquery language (Tang and Mooney, 2001). The average sentence length is 7.48, much shorter than RoboCup. 250 of the queries were also translated into Spanish, Japanese and Turkish, resulting in a smaller, multilingual data set.

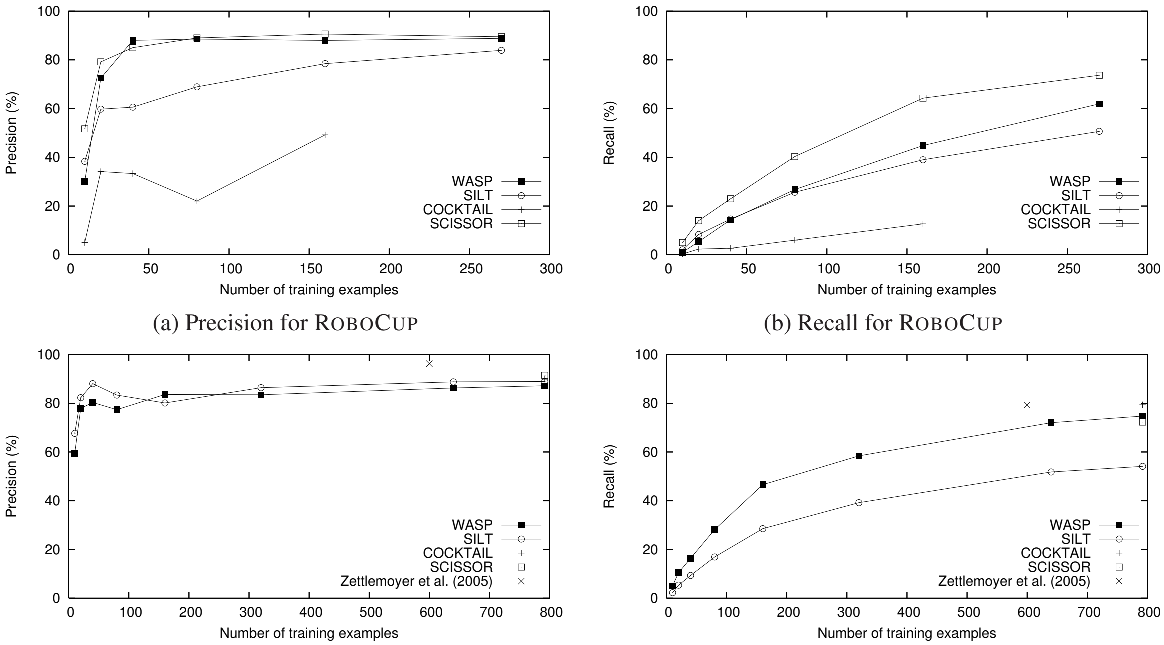
For each domain, there was a minimal set of *initial rules* representing knowledge needed for translating basic domain entities. These rules were always included in a lexicon. For example, in Geoquery, the initial rules were: Num $\rightarrow \langle x, x \rangle$, for all $x \in \mathbb{R}$; City $\rightarrow \langle c, \texttt{cityid('}c\texttt{',\_)} \rangle$, for all city names c (e.g. *new york*); and similar rules for other types of names (e.g. rivers). Name translations were provided for the multilingual data set (e.g. City $\rightarrow \langle$*nyuu yooku*, $\texttt{cityid('new york',\_)} \rangle$ for Japanese).

Standard 10-fold cross validation was used in our experiments. A semantic parser was learned from the training set. Then the learned parser was used to translate the test sentences into MRs. Translation failed when there were constructs that the parser did not cover. We counted the number of sentences that were translated into an MR, and the number of translations that were correct. For RoboCup, a translation was correct if it exactly matched the correct MR. For Geoquery, a translation was correct if it retrieved the same answer as the correct query. Using these counts, we measured the performance of the parser in terms of *precision* (percentage of translations that were correct) and *recall* (percentage of test sentences that were correctly translated). For RoboCup, it took 47 minutes to learn a parser using IIS. For Geoquery, it took 83 minutes.

Figure 6 shows the performance of Wasp compared to four other algorithms: Silt (Kate et al., 2005), Cocktail (Tang and Mooney, 2001), Scissor (Ge and Mooney, 2005) and Zettlemoyer and Collins (2005). Experimental results clearly show the advantage of extra supervision in Scissor and Zettlemoyer and Collins's parser (see Section 1). However, Wasp performs quite favorably compared to Silt and Cocktail, which use the same training data. In particular, Cocktail, a deterministic shift-reduce parser based on inductive logic programming, fails to scale up to the RoboCup domain where sentences are much longer, and crashes on larger training sets due to memory overflow. Wasp also outperforms Silt in terms of recall, where lexical learning is done by a local bottom-up search, which is much less effective than the word-alignment-based algorithm in Wasp.

Figure 7 shows the performance of Wasp on the multilingual Geoquery data set. The languages being considered differ in terms of word order: Subject-Verb-Object for English and Spanish, and Subject-Object-Verb for Japanese and Turkish. Wasp's performance is consistent across these languages despite some slight differences, most probably due to factors other than word order (e.g. lower recall for Turkish due to a much larger vocabulary). Details can be found in a longer version of this paper (Wong, 2005).

[1] We also implemented limited-memory BFGS (Nocedal, 1980). Preliminary experiments showed that it typically reduces training time by more than half with similar accuracy.



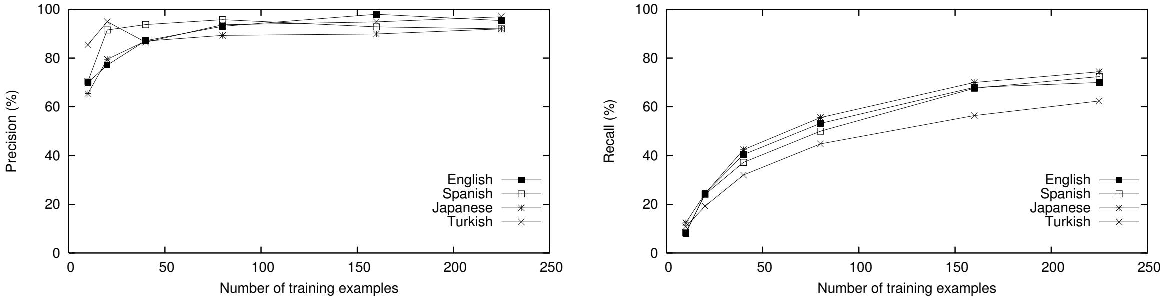
(a) Precision for RoboCup

(b) Recall for RoboCup

(c) Precision for GeoQuery

(d) Recall for GeoQuery

Figure 6: Precision and recall learning curves comparing various semantic parsers



(a) Precision for GeoQuery

(b) Recall for GeoQuery

Figure 7: Precision and recall learning curves comparing various natural languages

7 Conclusion

We have presented a novel statistical approach to semantic parsing in which a word-based alignment model is used for lexical learning, and the parsing model itself can be seen as a syntax-based translation model. Our method is like many phrase-based translation models, which require a simpler, word-based alignment model for the acquisition of a phrasal lexicon (Och and Ney, 2003). It is also similar to the hierarchical phrase-based model of Chiang (2005), in which hierarchical phrase pairs, essentially SCFG rules, are learned through the use of a simpler, phrase-based alignment model. Our work shows that ideas from compiler theory (SCFG) and machine translation (word alignment models) can be successfully applied to semantic parsing, a closely-related task whose goal is to translate a natural language into a formal language.

Lexical learning requires word alignments that are phrasally coherent. We presented a simple greedy algorithm for removing links that destroy phrasal coherence. Although it is shown to be quite effective in the current domains, it is preferable to have a more principled way of *promoting* phrasal coherence. The problem is that, by treating MRL productions as atomic units, current word-based alignment models have *no* knowledge about the tree structure hidden in a linearized MR parse. In the future, we would like to develop a word-based alignment model that is aware of the MRL syntax, so that better lexicons can be learned.

Acknowledgments

This research was supported by Defense Advanced Research Projects Agency under grant HR0011-04-1-0007.

References

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

J. F. Allen. 1995. *Natural Language Understanding (2nd Ed.)*. Benjamin/Cummings, Menlo Park, CA.

I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. 1995. Natural language interfaces to databases: An introduction. *Journal of Natural Language Engineering*, 1(1):29–81.

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

S. Chen and R. Rosenfeld. 1999. A Gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University, Pittsburgh, PA.

M. Chen et al. 2003. Users manual: RoboCup soccer server manual for soccer server version 7.07 and later. Available at `http://sourceforge.net/projects/sserver/`.

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL-05*, pages 263–270, Ann Arbor, MI, June.

R. Ge and R. J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proc. of CoNLL-05*, pages 9–16, Ann Arbor, MI, July.

R. J. Kate, Y. W. Wong, and R. J. Mooney. 2005. Learning to transform natural to formal languages. In *Proc. of AAAI-05*, pages 1062–1068, Pittsburgh, PA, July.

G. Kuhlmann, P. Stone, R. J. Mooney, and J. W. Shavlik. 2004. Guiding a reinforcement learner with natural language advice: Initial results in RoboCup soccer. In *Proc. of the AAAI-04 Workshop on Supervisory Control of Learning and Adaptive Systems*, San Jose, CA, July.

K. Macherey, F. J. Och, and H. Ney. 2001. Natural language understanding using statistical machine translation. In *Proc. of EuroSpeech-01*, pages 2205–2208, Aalborg, Denmark.

I. D. Melamed. 2004. Statistical machine translation by parsing. In *Proc. of ACL-04*, pages 653–660, Barcelona, Spain.

S. Miller, D. Stallard, R. Bobrow, and R. Schwartz. 1996. A fully statistical approach to natural language interfaces. In *Proc. of ACL-96*, pages 55–61, Santa Cruz, CA.

Y. Miyao and J. Tsujii. 2002. Maximum entropy estimation for feature forests. In *Proc. of HLT-02*, San Diego, CA, March.

J. Nocedal. 1980. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, July.

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

K. A. Papineni, S. Roukos, and R. T. Ward. 1997. Feature-based language understanding. In *Proc. of EuroSpeech-97*, pages 1435–1438, Rhodes, Greece.

S. Riezler, D. Prescher, J. Kuhn, and M. Johnson. 2000. Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training. In *Proc. of ACL-00*, pages 480–487, Hong Kong.

A. Stolcke. 1995. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201.

L. R. Tang and R. J. Mooney. 2001. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Proc. of ECML-01*, pages 466–477, Freiburg, Germany.

Y. W. Wong. 2005. Learning for semantic parsing using statistical machine translation techniques. Technical Report UT-AI-05-323, Artificial Intelligence Lab, University of Texas at Austin, Austin, TX, October.

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *Proc. of ACL-01*, pages 523–530, Toulouse, France.

J. M. Zelle and R. J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proc. of AAAI-96*, pages 1050–1055, Portland, OR, August.

L. S. Zettlemoyer and M. Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proc. of UAI-05*, Edinburgh, Scotland, July.