In *Proceedings of the COLING/ACL-2006 Main Conference Poster Sessions*. pp. 263-270, Sydney, Australia, July 2006.

# Discriminative Reranking for Semantic Parsing

**Ruifang Ge Raymond J. Mooney**
Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712
{grf,mooney}@cs.utexas.edu

## Abstract

Semantic parsing is the task of mapping natural language sentences to complete formal meaning representations. The performance of semantic parsing can be potentially improved by using discriminative reranking, which explores arbitrary global features. In this paper, we investigate discriminative reranking upon a baseline semantic parser, SCISSOR, where the composition of meaning representations is guided by syntax. We examine if features used for syntactic parsing can be adapted for semantic parsing by creating similar semantic features based on the mapping between syntax and semantics. We report experimental results on two real applications, an interpreter for coaching instructions in robotic soccer and a natural-language database interface. The results show that reranking can improve the performance on the coaching interpreter, but not on the database interface.

## 1 Introduction

A long-standing challenge within natural language processing has been to understand the meaning of natural language sentences. In comparison with shallow semantic analysis tasks, such as word-sense disambiguation (Ide and Jeanéronis, 1998) and semantic role labeling (Gildea and Jurafsky, 2002; Carreras and Màrquez, 2005), which only partially tackle this problem by identifying the meanings of target words or finding semantic roles of predicates, semantic parsing (Kate et al., 2005; Ge and Mooney, 2005; Zettlemoyer and Collins, 2005) pursues a more ambitious goal – mapping natural language sentences to complete formal meaning representations (MRs), where the meaning of each part of a sentence is analyzed, including noun phrases, verb phrases, negation, quantifiers and so on. Semantic parsing enables logic reasoning and is critical in many practical tasks, such as speech understanding (Zue and Glass, 2000), question answering (Lev et al., 2004) and advice taking (Kuhlmann et al., 2004).

Ge and Mooney (2005) introduced an approach, SCISSOR, where the composition of meaning representations is guided by syntax. First, a statistical parser is used to generate a semantically-augmented parse tree (SAPT), where each internal node includes both a syntactic and semantic label. Once a SAPT is generated, an additional meaning-composition process guided by the tree structure is used to translate it into a final formal meaning representation.

The performance of semantic parsing can be potentially improved by using discriminative reranking, which explores arbitrary global features. While reranking has benefited many tagging and parsing tasks (Collins, 2000; Collins, 2002c; Charniak and Johnson, 2005) including semantic role labeling (Toutanova et al., 2005), it has not yet been applied to semantic parsing. In this paper, we investigate the effect of discriminative reranking to semantic parsing.

We examine if the features used in reranking syntactic parses can be adapted for semantic parsing, more concretely, for reranking the top SAPTs from the baseline model SCISSOR. The syntactic features introduced by Collins (2000) for syntactic parsing are extended with similar semantic features, based on the coupling of syntax and semantics. We present experimental results on two corpora: an interpreter for coaching instructions in robotic soccer (CLANG) and a natural-language database interface (GeoQuery). The best reranking model significantly improves F-measure on CLANG from 82.3% to 85.1% (15.8% relative error reduction), however, it fails to show improvements on GEOQUERY.

## 2 Background

### 2.1 Application Domains

#### 2.1.1 CLANG: the RoboCup Coach Language

RoboCup (`www.robocup.org`) is an international AI research initiative using robotic soccer as its primary domain. In the Coach Competition, teams of agents compete on a simulated soccer field and receive advice from a team coach in a formal language called CLANG. In CLANG, tactics and behaviors are expressed in terms of if-then rules. As described in Chen et al. (2003), its grammar consists of 37 non-terminal symbols and 133 productions. Negation and quantifiers like *all* are included in the language. Below is a sample rule with its English gloss:

```
((bpos (penalty-area our))
 (do (player-except our {4})
     (pos (half our))))
```

*"If the ball is in our penalty area, all our players except player 4 should stay in our half."*

#### 2.1.2 GEOQUERY: a DB Query Language

GEOQUERY is a logical query language for a small database of U.S. geography containing about 800 facts. The GEOQUERY language consists of Prolog queries augmented with several meta-predicates (Zelle and Mooney, 1996). Negation and quantifiers like *all* and *each* are included in the language. Below is a sample query with its English gloss:

```
answer(A,count(B,(city(B),loc(B,C),
        const(C,countryid(usa))),A))
```

*"How many cities are there in the US?"*

### 2.2 SCISSOR: the Baseline Model

SCISSOR is based on a fairly standard approach to compositional semantics (Jurafsky and Martin, 2000). First, a statistical parser is used to construct a semantically-augmented parse tree that captures the semantic interpretation of individual words and the basic predicate-argument structure of a sentence. Next, a recursive deterministic procedure is used to compose the MR of a parent node from the MR of its children following the tree structure.

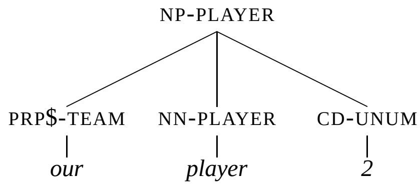


Figure 1: A SAPT for describing a simple CLANG concept PLAYER .

Figure 1 shows the SAPT for a simple natural language phrase describing the concept PLAYER in CLANG. We can see that each internal node in the parse tree is annotated with a semantic label (shown after dashes) representing *concepts* in an application domain; when a node is semantically vacuous in the application domain, it is assigned with the semantic label NULL. The semantic labels on words and non-terminal nodes represent the meanings of these words and constituents respectively. For example, the word *our* represents a TEAM concept in CLANG with the value *our*, whereas the constituent OUR PLAYER 2 represents a PLAYER concept. Some *type concepts* do not take arguments, like *team* and *unum* (uniform number), while some concepts, which we refer to as *predicates*, take an ordered list of arguments, like *player* which requires both a TEAM and a UNUM as its arguments.

SAPTs are given to a meaning composition process to compose meaning, guided by both tree structures and domain predicate-argument requirements. In figure 1, the MR of *our* and *2* would fill the arguments of PLAYER to generate the MR of the whole constituent PLAYER(OUR,2) using this process.

SCISSOR is implemented by augmenting Collins' (1997) head-driven parsing model II to incorporate the generation of semantic labels on internal nodes. In a head-driven parsing model, a tree can be seen as generated by expanding non-terminals with grammar rules recursively. To deal with the sparse data problem, the expansion of a non-terminal (parent) is decomposed into primitive steps: a child is chosen as the head and is generated first, and then the other children (modifiers) are generated independently

| BACK-OFFLEVEL | $\mathcal{P}_{L1}(L_i\|...)$ |
|---|---|
| 1 | P,H,$w$,$t$,$\Delta$,LC |
| 2 | P,H,$t$,$\Delta$,LC |
| 3 | P,H,$\Delta$,LC |
| 4 | P,H |
| 5 | P |

Table 1: Extended back-off levels for the semantic parameter $\mathcal{P}_{L1}(L_i|...)$, using the same notation as in Ge and Mooney (2005). The symbols $P$, $H$ and $L_i$ are the semantic label of the parent , head, and the *i*th left child, $w$ is the head word of the parent, $t$ is the semantic label of the head word, $\delta$ is the distance between the head and the modifier, and $LC$ is the left semantic subcat.

constrained by the head. Here, we only describe changes made to SCISSOR for reranking, for a full description of SCISSOR see Ge and Mooney (2005).

In SCISSOR, the generation of semantic labels on modifiers are constrained by semantic subcategorization frames, for which data can be very sparse. An example of a semantic subcat in Figure 1 is that the head PLAYER associated with NN requires a TEAM as its modifier. Although this constraint improves SCISSOR's precision, which is important for semantic parsing, it also limits its recall. To generate plenty of candidate SAPTs for reranking, we extended the back-off levels for the parameters generating semantic labels of modifiers. The new set is shown in Table 1 using the parameters for the generation of the left-side modifiers as an example. The back-off levels 4 and 5 are newly added by removing the constraints from the semantic subcat. Although the best SAPTs found by the model may not be as precise as before, we expect that reranking can improve the results and rank correct SAPTs higher.

### 2.3 The Averaged Perceptron Reranking Model

Averaged perceptron (Collins, 2002a) has been successfully applied to several tagging and parsing reranking tasks (Collins, 2002c; Collins, 2002a), and in this paper, we employed it in reranking semantic parses generated by the base semantic parser SCISSOR. The model is composed of three parts (Collins, 2002a): a set of candidate SAPTs $GEN$, which is the top $n$ SAPTs of a sentence from SCISSOR; a function $\Phi$ that maps a sentence

**Inputs:** A set of training examples $(x_i, y_i^*)$, $i = 1...n$, where $x_i$ is a sentence, and $y_i^*$ is a candidate SAPT that has the highest similarity score with the gold-standard SAPT
**Initialization:** Set $\bar{W} = 0$
**Algorithm:**
For $t = 1...T, i = 1...n$
Calculate $y_i = \arg\max_{y \in GEN(x_i)} \Phi(x_i, y) \cdot \bar{W}$
If $(y_i \neq y_i^*)$ then $\bar{W} = \bar{W} + \Phi(x_i, y_i^*) - \Phi(x_i, y_i)$
**Output:** The parameter vector $\bar{W}$

Figure 2: The perceptron training algorithm.

$x$ and its SAPT $y$ into a feature vector $\Phi(x, y) \in \mathbb{R}^d$; and a weight vector $\bar{W}$ associated with the set of features. Each feature in a feature vector is a function on a SAPT that maps the SAPT to a real value. The SAPT with the highest score under a parameter vector $\bar{W}$ is outputted, where the score is calculated as:

$$score(x, y) = \Phi(x, y) \cdot \bar{W} \tag{1}$$

The perceptron training algorithm for estimating the parameter vector $\bar{W}$ is shown in Figure 2. For a full description of the algorithm, see (Collins, 2002a). The averaged perceptron, a variant of the perceptron algorithm is often used in testing to decrease generalization errors on unseen test examples, where the parameter vectors used in testing is the average of each parameter vector generated during the training process.

## 3 Features for Reranking SAPTs

In our setting, reranking models discriminate between SAPTs that can lead to correct MRs and those that can not. Intuitively, both syntactic and semantic features describing the syntactic and semantic substructures of a SAPT would be good indicators of the SAPT's correctness.

The syntactic features introduced by Collins (2000) for reranking syntactic parse trees have been proven successfully in both English and Spanish (Cowan and Collins, 2005). We examine if these syntactic features can be adapted for semantic parsing by creating similar semantic features. In the following section, we first briefly describe the syntactic features introduced by Collins (2000), and then introduce two adapted semantic feature sets. A SAPT in CLANG is shown in Figure 3 for illustrating the features throughout this section.

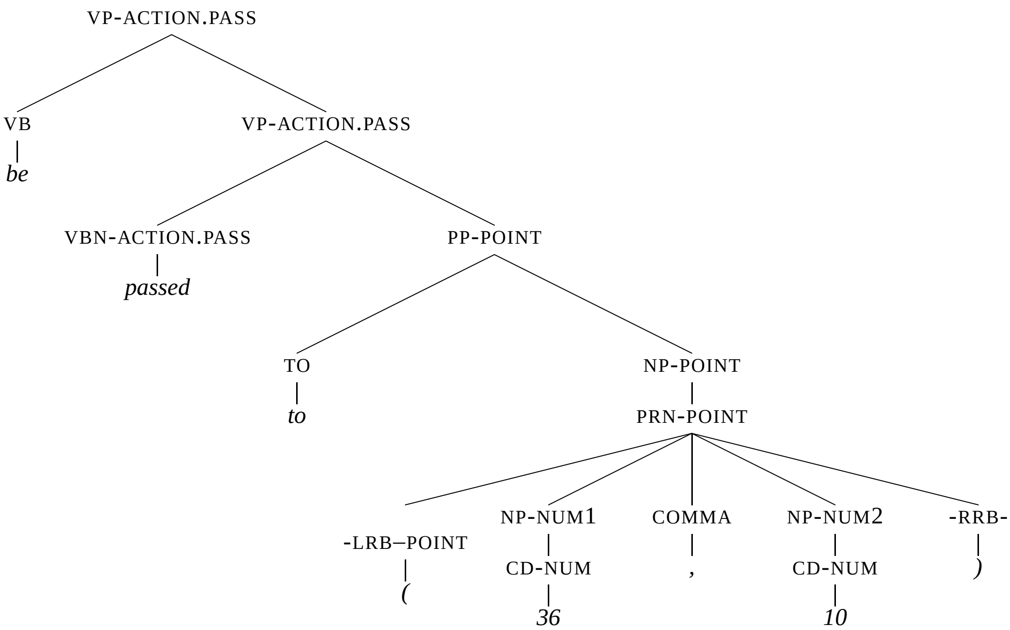


Figure 3: A SAPT for illustrating the reranking features, where the syntactic label "," is replaced by COMMA for a clearer description of features, and the NULL semantic labels are not shown. The head of the rule "PRN-POINT→ -LRB–POINT NP-NUM1 COMMA NP-NUM2 -RRB-" is -LRB–POINT. The semantic labels NUM1 and NUM2 are meta concepts in CLANG specifying the semantic role filled since NUM can fill multiple semantic roles in the predicate POINT.

### 3.1 Syntactic Features

All syntactic features introduced by Collins (2000) are included for reranking SAPTs. While the full description of all the features is beyond the scope of this paper, we still introduce several feature types here for the convenience of introducing semantic features later.

1. Rules. These are the counts of unique syntactic context-free rules in a SAPT. The example in Figure 3 has the feature $f$(PRN→ -LRB- NP COMMA NP -RRB-)=1.
2. Bigrams. These are the counts of unique bigrams of syntactic labels in a constituent. They are also featured with the syntactic label of the constituent, and the bigram's relative direction (*left*, *right*) to the head of the constituent. The example in Figure 3 has the feature $f$(NP COMMA, *right*, PRN)=1.
3. Grandparent Rules. These are the same as Rules, but also include the syntactic label above a rule. The example in Figure 3 has the feature $f$([PRN→ -LRB- NP COMMA NP -RRB-], NP)=1, where NP is the syntactic label above the rule "PRN→ -LRB- NP COMMA NP -RRB-".
4. Grandparent Bigrams. These are the same as Bigrams, but also include the syntactic label above the constituent containing a bigram. The example in Figure 3 has the feature $f$([NP COMMA, *right*, PRN], NP)=1, where NP is the syntactic label above the constituent PRN.

### 3.2 Semantic Features

#### 3.2.1 Semantic Feature Set I

A similar semantic feature type is introduced for each syntactic feature type used by Collins (2000) by replacing syntactic labels with semantic ones (with the semantic label NULL not included). The corresponding semantic feature types for the features in Section 3.1 are:

1. Rules. The example in Figure 3 has the feature $f$(POINT→ POINT NUM1 NUM2)=1.
2. Bigrams. The example in Figure 3 has the feature $f$(NUM1 NUM2, *right*, POINT)=1, where the bigram "NUM1 NUM2"appears to the right of the head POINT.
3. Grandparent Rules. The example in Figure 3 has the feature $f$([POINT→ POINT NUM1 NUM2], POINT)=1, where the last POINT is

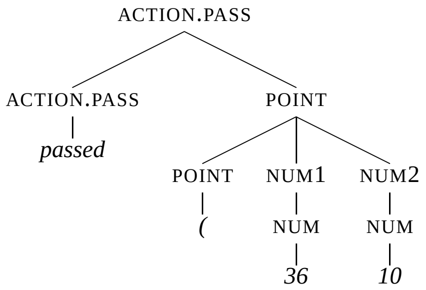


Figure 4: The tree generated by removing purely-syntactic nodes from the SAPT in Figure 3 (with syntactic labels omitted.)

the semantic label above the semantic rule "POINT→ POINT NUM1 NUM2".

4. Grandparent Bigrams. The example in Figure 3 has the feature $f$([NUM1 NUM2, *right*, POINT], POINT)=1, where the last POINT is the semantic label above the POINT associated with PRN.

#### 3.2.2 Semantic Feature Set II

Purely-syntactic structures in SAPTs exist with no meaning composition involved, such as the expansions from NP to PRN, and from PP to "TO NP" in Figure 3. One possible drawback of the semantic features derived directly from SAPTs as in Section 3.2.1 is that they could include features with no meaning composition involved, which are intuitively not very useful. For example, the nodes with purely-syntactic expansions mentioned above would trigger a semantic rule feature with meaning unchanged (from POINT to POINT). Another possible drawback of these features is that the features covering broader context could potentially fail to capture the real high-level meaning composition information. For example, the Grandparent Rule example in Section 3.2.1 has POINT as the semantic grandparent of a POINT composition, but not the real one ACTION.PASS.

To address these problems, another semantic feature set is introduced by deriving semantic features from trees where purely-syntactic nodes of SAPTs are removed (the resulting tree for the SAPT in Figure 3 is shown in Figure 4). In this tree representation, the example in Figure 4 would have the Grandparent Rule feature $f$([POINT→ POINT NUM1 NUM2], ACTION.PASS)=1, with the correct semantic grandparent ACTION.PASS included.

## 4 Experimental Evaluation

### 4.1 Experimental Methodology

Two corpora of natural language sentences paired with MRs were used in the reranking experiments. For CLANG, 300 pieces of coaching advice were randomly selected from the log files of the 2003 RoboCup Coach Competition. Each formal instruction was translated into English by one of four annotators (Kate et al., 2005). The average length of an natural language sentence in this corpus is 22.52 words. For GEOQUERY, 250 questions were collected by asking undergraduate students to generate English queries for the given database. Queries were then manually translated into logical form (Zelle and Mooney, 1996). The average length of a natural language sentence in this corpus is 6.87 words.

We adopted standard 10-fold cross validation for evaluation: 9/10 of the whole dataset was used for training (training set), and 1/10 for testing (test set). To train a reranking model on a training set, a separate "internal" 10-fold cross validation over the training set was employed to generate $n$-best SAPTs for each training example using a baseline learner, where each training set was again separated into 10 folds with 9/10 for training the baseline learner, and 1/10 for producing the $n$-best SAPTs for training the reranker. Reranking models trained in this way ensure that the $n$-best SAPTs for each training example are not generated by a baseline model that has already seen that example. To test a reranking model on a test set, a baseline model trained on a whole training set was used to generate $n$-best SAPTs for each test example, and then the reranking model trained with the above method was used to choose a best SAPT from the candidate SAPTs.

The performance of semantic parsing was measured in terms of *precision* (the percentage of completed MRs that were correct), *recall* (the percentage of all sentences whose MRs were correctly generated) and F-measure (the harmonic mean of precision and recall). Since even a single mistake in an MR could totally change the meaning of an example (e.g. having OUR in an MR instead of OPPONENT in CLANG), no partial credit was given for examples with partially-correct SAPTs.

Averaged perceptron (Collins, 2002a), which has been successfully applied to several tagging and parsing reranking tasks (Collins, 2002c; Collins, 2002a), was employed for training rerank-

| | CLANG | | | GEOQUERY | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| SCISSOR | **89.5** | 73.7 | 80.8 | **98.5** | 74.4 | 84.8 |
| SCISSOR+ | 87.0 | **78.0** | **82.3** | 95.5 | **77.2** | **85.4** |

Table 2: The performance of the baseline model SCISSOR+ compared with SCISSOR (with the best result in bold), where $P$ = precision, $R$ = recall, and $F$ = F-measure.

| *n* | 1 | 2 | 5 | 10 | 20 | 50 |
|---|---|---|---|---|---|---|
| CLANG | 78.0 | 81.3 | 83.0 | 84.0 | 85.0 | 85.3 |
| GEOQUERY | 77.2 | 77.6 | 80.0 | 81.2 | 81.6 | 81.6 |

Table 3: Oracle recalls on CLANG and GEOQUERY as a function of number *n* of *n*-best SAPTs.

ing models. To choose the correct SAPT of a training example required for training the averaged perceptron, we selected a SAPT that results in the correct MR; if multiple such SAPTs exist, the one with the highest baseline score was chosen. Since no partial credit was awarded in evaluation, a training example was discarded if it had no correct SAPT. Rerankers were trained on the 50-best SAPTs provided by SCISSOR, and the number of perceptron iterations over the training examples was limited to 10. Typically, in order to avoid over-fitting, reranking features are filtered by removing those occurring in less than some minimal number of training examples. We only removed features that never occurred in the training data since experiments with higher cut-offs failed to show any improvements.

### 4.2 Results

#### 4.2.1 Baseline Results

Table 2 shows the results comparing the baseline learner SCISSOR using both the back-off parameters in Ge and Mooney (2005) (SCISSOR) and the revised parameters in Section 2.2 (SCISSOR+). As we expected, SCISSOR+ has better recall and worse precision than SCISSOR on both corpora due to the additional levels of back-off. SCISSOR+ is used as the baseline model for all reranking experiments in the next section.

Table 3 gives oracle recalls for CLANG and GEOQUERY where an oracle picks the correct parse from the $n$-best SAPTs if *any* of them are correct. Results are shown for increasing values of $n$. The trends for CLANG and GEOQUERY are different: small values of $n$ show significant improvements for CLANG, while a larger $n$ is needed to improve results for GEOQUERY.

#### 4.2.2 Reranking Results

In this section, we describe the experiments with reranking models utilizing different feature sets. All models include the score assigned to a SAPT by the baseline model as a special feature.

Table 4 shows results using different feature sets derived directly from SAPTs. In general, reranking improves the performance of semantic parsing on CLANG, but not on GEOQUERY. This could be explained by the different oracle recall trends of CLANG and GEOQUERY. We can see that in Table 3, even a small $n$ can increase the oracle score on CLANG significantly, but not on GEOQUERY. With the baseline score included as a feature, correct SAPTs closer to the top are more likely to be reranked to the top than the ones in the back, thus CLANG is more likely to have more sentences reranked correct than GEOQUERY. On CLANG, using the semantic feature set alone achieves the best improvements over the baseline with 2.8% absolute improvement in F-measure (15.8% relative error reduction), which is significant at the 95% confidence level using a paired Student's t-test. Nevertheless, the difference between $\text{SEM}_1$ and $\text{SYN+SEM}_1$ is very small (only one example). Using syntactic features alone only slightly improves the results because the syntactic features do not directly discriminate between correct and incorrect meaning representations. To put this in perspective, Charniak and Johnson (2005) reported that reranking improves the F-measure of syntactic parsing from 89.7% to 91.0% with a *50*-best oracle F-measure score of 96.8%.

Table 5 compares results using semantic features directly derived from SAPTs ($\text{SEM}_1$), and from trees with purely-syntactic nodes removed ($\text{SEM}_2$). It compares reranking models using these

| | CLANG | | | GEOQUERY | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| SCISSOR+ | 87.0 | 78.0 | 82.3 | **95.5** | **77.2** | **85.4** |
| SYN | 87.7 | 78.7 | 83.0 | **95.5** | **77.2** | **85.4** |
| $\text{SEM}_1$ | **90.0(23.1)** | **80.7(12.3)** | **85.1(15.8)** | 95.5 | 76.8 | 85.1 |
| $\text{SYN+SEM}_1$ | 89.6 | 80.3 | 84.7 | 95.5 | 76.4 | 84.9 |

Table 4: Reranking results on CLANG and GEOQUERY using different feature sets derived directly from SAPTs (with the best results in bold and relative error reduction in parentheses). The reranking model SYN uses the syntactic feature set in Section 3.1, $\text{SEM}_1$ uses the semantic feature set in Section 3.2.1, and $\text{SYN+SEM}_1$ uses both.

| | CLANG | | | GEOQUERY | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| $\text{SEM}_1$ | **90.0** | **80.7** | **85.1** | 95.5 | 76.8 | 85.1 |
| $\text{SEM}_2$ | 88.1 | 79.0 | 83.3 | **96.0** | **77.2** | **85.6** |
| $\text{SEM}_1\text{+SEM}_2$ | 88.5 | 79.3 | 83.7 | 95.5 | 76.4 | 84.9 |
| $\text{SYN+SEM}_1$ | 89.6 | 80.3 | 84.7 | 95.5 | 76.4 | 84.9 |
| $\text{SYN+SEM}_2$ | 88.1 | 79.0 | 83.3 | 95.5 | 76.8 | 85.1 |
| $\text{SYN+SEM}_1\text{+SEM}_2$ | 88.9 | 79.7 | 84.0 | 95.5 | 76.4 | 84.9 |

Table 5: Reranking results on CLANG and GEOQUERY comparing semantic features derived directly from SAPTs, and semantic features from trees with purely-syntactic nodes removed. The symbol $\text{SEM}_1$ and $\text{SEM}_2$ refer to the semantic feature sets in Section 3.2.1 and 3.2.1 respectively, and SYN refers to the syntactic feature set in Section 3.1.

feature sets alone and together, and using them along with the syntactic feature set (SYN) alone and together. Overall, $\text{SEM}_1$ provides better results than $\text{SEM}_2$ on CLANG and slightly worse results on GEOQUERY (only in one sentence), regardless of whether or not syntactic features are included. Using both semantic feature sets does not improve the results over just using $\text{SEM}_1$. On one hand, the better performance of $\text{SEM}_1$ on CLANG contradicts our expectation because of the reasons discussed in Section 3.2.2; the reason behind this needs to be investigated. On the other hand, however, it also suggests that the semantic features derived directly from SAPTs can provide good evidence for semantic correctness, even with redundant purely syntactically motivated features.

We have also informally experimented with smoothed semantic features utilizing domain ontology given by CLANG, which did not show improvements over reranking models not using these features.

## 5 Conclusion

We have applied discriminative reranking to semantic parsing, where reranking features are developed from features for reranking syntactic parses based on the coupling of syntax and semantics. The best reranking model significantly improves F-measure on a Robocup coaching task (CLANG) from 82.3% to 85.1%, while it fails to improve the performance on a geography database query task (GEOQUERY).

Future work includes further investigation of the reasons behind the different utility of reranking for the CLANG and GEOQUERY tasks. We also plan to explore other types of reranking features, such as the features used in semantic role labeling (SRL) (Gildea and Jurafsky, 2002; Carreras and Màrquez, 2005), like the path between a target predicate and its argument, and kernel methods (Collins, 2002b). Experimenting with other effective reranking algorithms, such as SVMs (Joachims, 2002) and MaxEnt (Charniak and Johnson, 2005), is also a direction of our future research.

## 6 Acknowledgements

We would like to thank Rohit J. Kate and anonymous reviewers for their insightful comments. This research was supported by Defense Advanced Research Projects Agency under grant HR0011-04-1-0007.

## References

Xavier Carreras and Luís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proc. of 9th Conf. on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, MI, June.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proc. of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 173–180, Ann Arbor, MI, June.

Mao Chen, Ehsan Foroughi, Fredrik Heintz, Spiros Kapetanakis, Kostas Kostiadis, Johan Kummeneje, Itsuki Noda, Oliver Obst, Patrick Riley, Timo Steffens, Yi Wang, and Xiang Yin. 2003. Users manual: RoboCup soccer server manual for soccer server version 7.07 and later. Available at `http://sourceforge.net/projects/sserver/`.

Michael J. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 16–23.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proc. of 17th Intl. Conf. on Machine Learning (ICML-2000)*, pages 175–182, Stanford, CA, June.

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

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

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

Brooke Cowan and Michael Collins. 2005. Morphology and reranking for the statistical parsing of Spanish. In *Proc. of the Human Language Technology Conf. and Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP-05)*, Vancouver, B.C., Canada, October.

Ruifang Ge and Raymond J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proc. of 9th Conf. on Computational Natural Language Learning (CoNLL-2005)*, pages 9–16, Ann Arbor, MI, July.

Daniel Gildea and Daniel Jurafsky. 2002. Automated labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Nancy A. Ide and Jeanéronis. 1998. Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1–40.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proc. of 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD-2002)*, Edmonton, Canada.

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, Y. W. Wong, and R. J. Mooney. 2005. Learning to transform natural to formal languages. In *Proc. of 20th Natl. Conf. on Artificial Intelligence (AAAI-2005)*, pages 1062–1068, Pittsburgh, PA, July.

Gregory Kuhlmann, Peter Stone, Raymond J. Mooney, and Jude 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.

Iddo Lev, Bill MacCartney, Christopher D. Manning, and Roger Levy. 2004. Solving logic puzzles: From robust processing to precise semantics. In *Proc. of 2nd Workshop on Text Meaning and Interpretation, ACL-04*, Barcelona, Spain.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2005. Joint learning improves semantic role labeling. In *Proc. of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, Ann Arbor, MI, June.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proc. of 13th Natl. Conf. 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 *Proc. of 21th Conf. on Uncertainty in Artificial Intelligence (UAI-2005)*, Edinburgh, Scotland, July.

Victor W. Zue and James R. Glass. 2000. Conversational interfaces: Advances and challenges. In *Proc. of the IEEE*, volume 88(8), pages 1166–1180.