Computational Linguistics and Intelligent Text Processing: Proceedings of the 8th International Conference, CICLing 2007, Mexico City (invited paper), A. Gelbukh (Ed.), pp. 311-324, Springer, Berlin, Germany, February 2007.

Learning for Semantic Parsing

Raymond J. Mooney

Department of Computer Sciences, University of Texas at Austin
1 University Station C0500, Austin, TX 78712-0233, USA
mooney@cs.utexas.edu

Abstract. Semantic parsing is the task of mapping a natural language sentence into a complete, formal meaning representation. Over the past decade, we have developed a number of machine learning methods for inducing semantic parsers by training on a corpus of sentences paired with their meaning representations in a specified formal language. We have demonstrated these methods on the automated construction of natural-language interfaces to databases and robot command languages. This paper reviews our prior work on this topic and discusses directions for future research.

1 Introduction

Semantic parsing is the task of mapping a natural language (NL) sentence into a complete, formal *meaning representation* (MR) or logical form. A *meaning representation language* (MRL) is a formal unambiguous language that allows for automated inference and processing, such as first-order predicate logic. In particular, our research has focused on applications in which the MRL is "executable" and can be directly used by another program to perform some task such as answering questions from a database or controlling the actions of a real or simulated robot. This distinguishes the task from related tasks such as semantic role labeling [8] and other forms of "shallow" semantic parsing which do not generate complete, formal representations.

Over the past decade, we have developed a number of systems for learning parsers that map NL sentences to a pre-specified MRL [44, 35, 37, 24, 17, 39, 23]. Given a training corpus of sentences annotated with their correct semantic interpretation in a given MRL, the goal of these systems is to induce an efficient and accurate semantic parser that can map novel sentences into this MRL. Some of the systems require extra training input in addition to (NL, MR) pairs, such as syntactic parse trees or semantically annotated parse trees.

In this paper, we first describe the applications we have explored and their corresponding MRLs, and then review the parsing and learning systems that we have already developed for these applications, along with experimental results on their performance. We then discuss important areas for future research in learning for semantic parsing.

2 Sample Applications and their MRLs

We have previously considered two MRLs for performing useful, complex tasks. The first is a database query language, primarily using a sample database on U.S. geography. The second MRL is a coaching language for robotic soccer developed for the RoboCup Coach Competition, in which AI researchers compete to provide effective instructions to a coachable team of agents in a simulated soccer domain [9].

When exploring NL interfaces for databases, the MRL we have primarily used is a logical query language based on Prolog. We have primarily focused on queries to a small database on U.S. geography. This domain, GEOQUERY, was originally chosen to test corpus-based semantic parsing due to the availability of a hand-built natural-language interface, GEOBASE, supplied with Turbo Prolog 2.0 [3]. The language consists of Prolog queries augmented with several meta-predicates [44]. Below is a sample query with its English gloss:

```
answer(A,count(B,(state(B),const(C,riverid(mississippi)),traverse(C,B)),A))
```

"How many states does the Mississippi run through?"

The same query language has also been used to build NLI's for databases of restaurants and CS-job openings, including a component that translates our logical queries to standard SQL database queries [36, 35]. The resulting formal queries can be executed to generate answers to the corresponding questions.

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 [9], its grammar consists of 37 non-terminal symbols and 133 productions. 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."

The robots in the simulator can interpret the CLANG instructions which then strongly affect their behavior while playing the game. The semantic parsers we have developed for this MRL were part of a larger research project on advice-taking reinforcement learners that can accept advice stated in natural language [25].

3 Systems for Learning Semantic Parsers

Our earliest system for learning semantic parsers called CHILL [44, 35] uses Inductive Logic Programming (ILP) [26] to learn a deterministic parser written in Prolog. In our more recent work, we have developed three different approaches

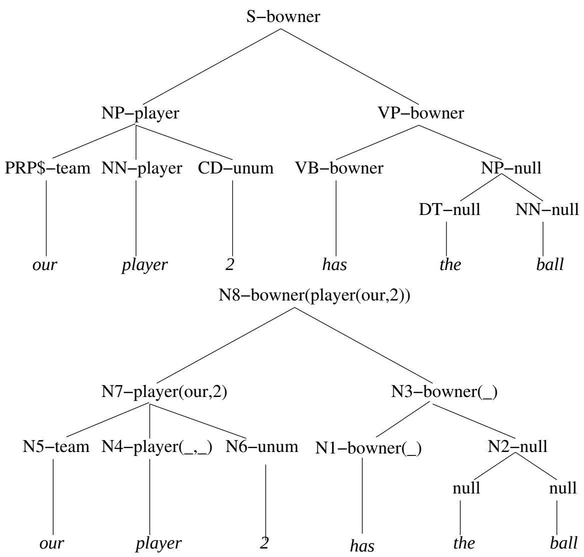


Fig. 1. The SAPT and its Compositional MR Construction for a CLang Sentence.

to learning statistical semantic parsers that are more robust and scale more effectively to larger training sets. Each exploits a different advanced technology in statistical natural language processing. Scissor [17, 18] adds detailed semantics to a state-of-the-art statistical syntactic parser (i.e. the Collins parser [12]), Wasp [39] adapts statistical machine translation methods to map from NL to MRL, and Krisp [23] uses Support Vector Machines (SVM's) [13] with a subsequence kernel specialized for text learning [27]. We briefly review each of these systems below. A version of our Geoquery data has also been used to evaluate a system for learning semantic parsers using probabilistic Combinatorial Categorial Grammars (CCG) [45].

3.1 Scissor

Scissor (Semantic Composition that Integrates Syntax and Semantics to get Optimal Representations) [17, 18] learns a statistical parser that generates a *semantically augmented parse tree* (SAPT), in which each internal node is given both a syntactic and a semantic label. We augment Collins' head-driven model 2 [12] to incorporate a semantic label on each internal node. By integrating syntactic and semantic interpretation into a single statistical model and finding the globally most probable parse, an accurate combined analysis can be obtained. Once an SAPT is generated, an additional step is required to translate it into a final MR.

In an SAPT, each internal node in the parse tree is annotated with a semantic label from the MRL. The left half of Fig. 1 shows the SAPT for a simple sentence in the CLang domain. The semantic labels (shown after the dashes) are *concepts* in the MRL. Some *type concepts* do not take arguments, like *team* and *unum* (uniform number). Some concepts, referred to as *predicates*, take an ordered list of arguments, like *player* and *bowner* (ball owner). Each predicate has a set of known semantic constraints on its arguments, specified in terms of concepts that can fill each argument, such as *player*(*team*, *unum*) and *bowner*(*player*). A special semantic label *null* is used for nodes that do not correspond to any concept in the domain. Training data for SCISSOR consists of (NL, SAPT, MR) triples.

First, an enhanced version of Collin's parser is trained to produce SAPTs instead of purely syntactic parse trees by adapting it to predict two labels for each node instead of one (see [17] for details). Next, a recursive procedure is used to compositionally construct the MR for each node in the SAPT given the MRs of its children. The right half of Fig. 1 illustrates the construction of the MR for the SAPT in the left half of the figure (nodes are numbered in the order in which the construction of their MRs are completed). In this process, semantic constraints are used to determine how to properly fill the arguments of a predicate for a node with the MRs of the node's children.

3.2 WASP

WASP (Word Alignment-based Semantic Parsing) [39] uses state-of-the-art Statistical Machine Translation (SMT) techniques [4, 5, 41, 10] to learn semantic parsers. SMT methods learn effective machine translation systems by training on *parallel corpora* consisting of human translations of documents into one or more alternative natural languages. The resulting translators are typically significantly more effective than manually developed systems and SMT has become the dominant approach to machine translation. We have adapted such methods to learn to translate from NL to MRL rather than from one NL to another.

WASP requires no prior knowledge of the NL syntax, although it assumes that an unambiguous, context-free grammar (CFG) of the target MRL is available. Since MRLs are formal computer-interpretable languages, such a grammar is usually easily available. First, an SMT word alignment system, GIZA++ [28, 5], is used to acquire a bilingual lexicon consisting of NL substrings coupled with their translations in the target MRL. As formal languages, MRLs frequently contain many purely syntactic tokens such as parentheses or brackets, which are difficult to align with words in NL. Consequently, we found it was much more effective to align words in the NL with productions of the MRL grammar used in the parse of the corresponding MR. Therefore, GIZA++ is used to produce an N to 1 alignment between the words in the NL sentence and a sequence of MRL productions corresponding to a top-down left-most derivation of the corresponding MR. A sample partial alignment is shown in Fig. 2.

Complete MRs are then formed by combining these NL substrings and their translations under a parsing framework called a synchronous CFG (SCFG) [1], which forms the basis of most existing statistical syntax-based translation models [41, 10]. In an SCFG, the right hand side of each production rule contains two strings, in our case one in NL and the other in MR. Derivations of the SCFG simultaneously produce NL sentences and their corresponding MRs. The bilingual lexicon acquired from word alignments on the training data is used to construct

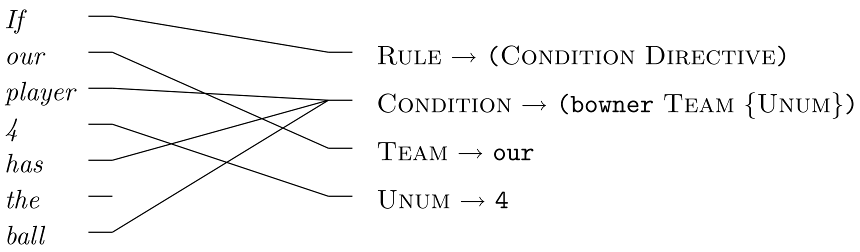


Fig. 2. Partial word alignment for the CLang statement and its English gloss

a set of SCFG production rules. A probabilistic parser is then produced by training a maximum-entropy model using expectation maximization (EM) to learn parameters for each of these SCFG productions, similar to the methods used in [30, 45]. To translate a novel NL sentence into its MR, a probabilistic chart parser [34] is used find the most probable synchronous derivation that generates the given NL, and the corresponding MR generated by this derivation is returned.

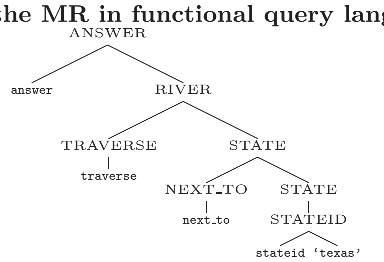
3.3 Krisp

Krisp (Kernel-based Robust Interpretation for Semantic Parsing) [23] uses SVMs with string kernels to build semantic parsers that are more robust in the presence of noisy training data. SVMs are state-of-the-art machine learning methods that learn maximum-margin separators to prevent over-fitting in very high-dimensional data such as natural language text [22]. They can be extended to non-linear separators and non-vector data by exploiting *kernels* that implicitly create an even higher dimensional space in which complex data is (nearly) linearly separable [32]. Recently, kernels over strings and trees have been effectively applied to a variety of problems in text learning and NLP [27, 43, 11, 6, 7]. In particular, Krisp uses the string kernel introduced in [27] to classify substrings in an NL sentence.

First, Krisp learns classifiers that recognize when a word or phrase in an NL sentence indicates that a particular concept in the MRL should be introduced into its MR. Like Wasp, it uses production rules in the MRL grammar to represent semantic concepts, and it learns classifiers for each production that classify NL substrings as indicative of that production or not. When semantically parsing a sentence, each classifier estimates the probability of each production covering different substrings of the sentence. This information is then used to compositionally build a complete MR for the sentence.

Krisp learns a semantic parser iteratively, each iteration improving upon the parser learned in the last iteration. In each iteration, for every production π in the MRL grammar, Krisp collects positive and negative examples. In the first iteration, the set of positive examples for production π contains all sentences whose MR parse tree uses the production π. The set of negative examples includes all of the other training sentences. Using these positive and negative examples, an SVM classifier[1] is trained for each production π using a string kernel. In subsequent iterations, the training examples are refined to more specific substrings within the sentences until the classifiers converge, analogous to iterations in EM [14].

NL: *"Which rivers run through the states bordering Texas?"*
Functional query language: `answer(traverse(next_to(stateid('texas'))))`
Parse tree of the MR in functional query language:



Productions:

ANSWER → `answer`(RIVER)	RIVER → TRAVERSE(STATE)
STATE → NEXT_TO(STATE)	STATE → STATEID
TRAVERSE → `traverse`	NEXT_TO → `next_to`
STATEID → `stateid('texas')`	

Fig. 3. An example of an NL query and its MR and its parse tree.

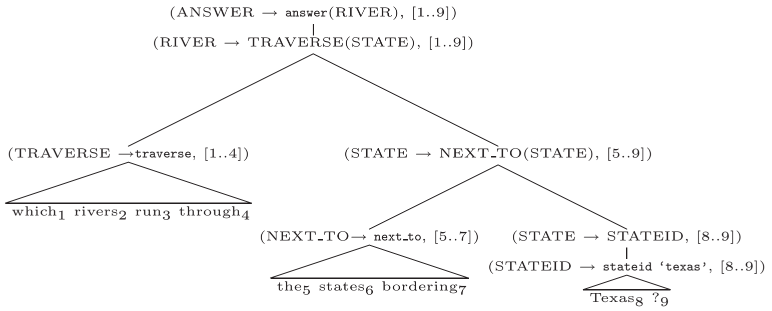


Fig. 4. Semantic derivation of the example in Fig. 3

During semantic parsing, KRISP uses these classifiers to find the most probable *semantic derivation* of a sentence. A *semantic derivation* of an NL sentence is a parse tree of an MR such that each node in the tree covers a substring of the sentence. The substrings covered by the children of a node are not allowed to overlap, and the substring covered by the parent must be the concatenation of the substrings covered by its children. Figure 4 shows a semantic derivation

[1] We use the LIBSVM package available at: `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

of the geography query and its MR parse shown in Fig. 3. The MRL used is a functional version of the formal database query language. The probability that a given production covers its corresponding substring is estimated using the SVM classifier for that production. Assuming independence, the probability of a semantic derivation is computed as the product of the probabilities for each of its productions. An adaptation of Earley's context-free parsing algorithm [15] is used to efficiently compute the most probable semantic derivation for a novel sentence, and this derivation directly determines its output MR.

4 Experimental Evaluation

Two corpora of NL sentences paired with MRs were used to evaluate our approaches. 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 [24]. The average length of an NL 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 [44]. The average length of an NL sentence in this corpus is 6.87 words. The queries in this corpus are more complex than those in the ATIS database-query corpus used in the speech community [46] which makes the Geoquery problem harder, as also shown by the results in [29].

Semantic-parser learning was evaluated using standard 10-fold cross validation. A given system may be unable to parse a particular sentence and therefore fail to produce an output MR. For each system, we measured the number of novel test sentences that resulted in complete MRs, and the number of these MRs that were correct. For CLang, an MR is correct iff it exactly matches the correct representation, up to reordering of the arguments of commutative operators like `and`. For Geoquery, an MR is correct iff the resulting query retrieved the same answer as the gold-standard MR when submitted to the database. The performance of each parser was then measured in terms of *precision* (the percentage of completed MRs that were correct) and *recall* (the percentage of all sentences with correctly generated MRs).

We used the version of Chill presented in [35], which uses the improved Cocktail ILP system and produces more accurate parsers than the original version presented in [44]. In the Geoquery domain, we also compare to the original hand-built semantic parser Geobase.

Figure 5 shows the precision and recall learning curves for Geoquery, and Fig. 6 shows similar results for CLang. Since Chill is very memory intensive, it could not be run with larger training sets from the more complex CLang corpus, where overall it does quite poorly. Although the precision of the commercial manually-developed system Geobase is fairly high, its recall is very low, illustrating the advantages of a learning approach. Overall, the three new learning systems do very well in both domains, learning quite accurate parsers after seeing a modest amount of training data. In general, Scissor gives the

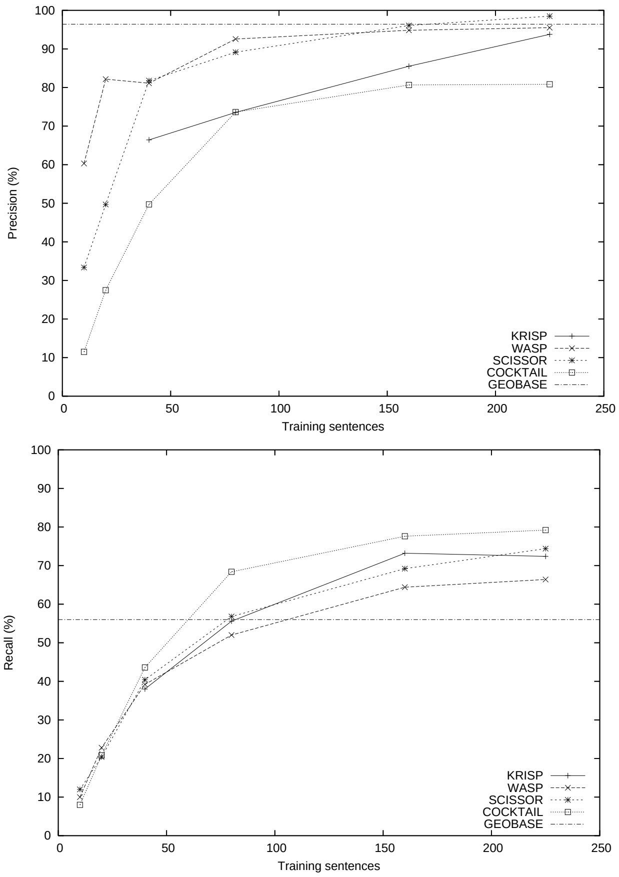
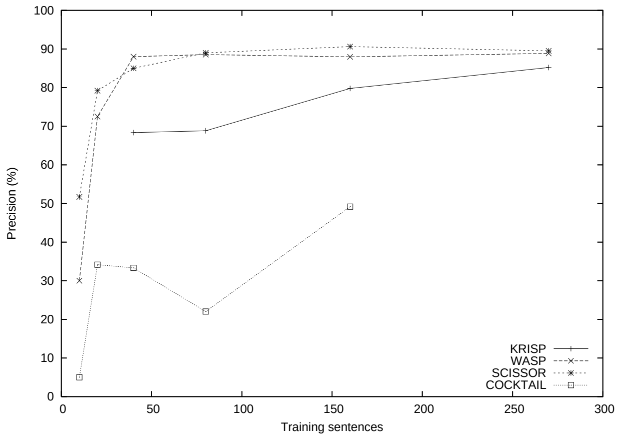


Fig. 5. Precision and Recall Learning Curves for GEOQUERY.

best results, but it also requires more detailed supervision in the form of SAPTs in addition to MRs. SCISSOR does particularly well on longer sentences where having a detailed traditional syntactic analysis helps in composing the correct MR. CHILL, which performs quite well in the GEOQUERY domain, does quite poorly on the longer, more complex sentences in the CLANG domain, where its local, deterministic decisions are less accurate.

The GEOQUERY corpus has also been translated into Spanish, Turkish, and Japanese. CHILL, KRISP, and WASP have also learned semantic parsers for these languages [37, 23, 39] and the accuracy results are similar to those shown above for English, demonstrating the generality of these approaches. SCISSOR has not been tested on this data since it requires additional supervision in the form of SAPTs, which are currently unavailable for these languages. Since KRISP relies on probabilistic string classifiers and does not require sentences to be parsable by a symbolic grammar, it is more robust to noisy input than the other systems. Experiments on artificially adding noise to sentences by simulating



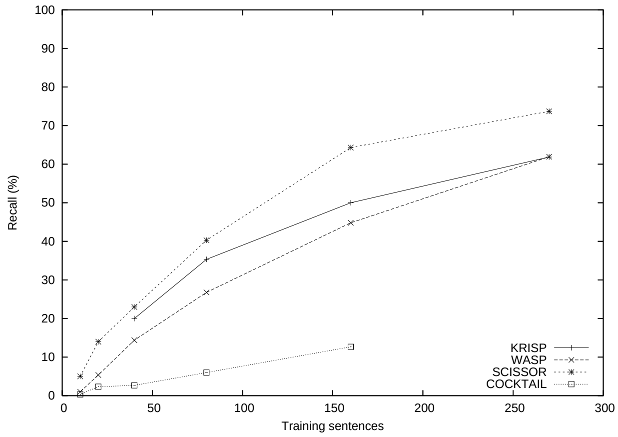


Fig. 6. Precision and Recall Learning Curves for Robocup CLang.

speech-recognition errors have demonstrated that Krisp's accuracy degrades less rapidly as more noise is added to the corpus [23].

5 Future Research

Although overall Scissor is somewhat more accurate in our current experimental results, this is not surprising since it requires additional human annotation in the form of SAPTs. Recent results have shown that Wasp and Krisp can also be adapted to benefit from the information in SAPTs; however, they still do not quite match the accuracy of Scissor. Therefore, we are currently exploring how, during training, the construction of SAPTs might be automated given only (NL MR) pairs and a general syntactic parser for the given natural language.

Currently, our evaluation of semantic parsers has been restricted to limited domains. This is largely due to the difficulty of developing an open-domain MRL and constructing a large annotated corpus of (NL MR) pairs for domain general text. As more general large corpora are developed that are annotated with deeper semantic representations, such as the OntoNotes corpus currently being assembled [21], we plan to test our systems on them. Given appropriately annotated large corpora, we are reasonably hopeful that our methods will scale to more general domains. A related direction of research is adapting our systems to learn "shallower" semantic parsers that produce incompletely formalized semantic representations such as those used in the FrameNet project [16].

However, we also believe there are many practical applications of domain-specific semantic parsers. Although the first NL database interfaces were developed in the 1970's [40, 38, 20], the technology was never successfully commercialized because of the significant manual software-engineering effort required to develop specialized systems for individual databases. We believe that by using learning techniques to automatically construct systems from annotated corpora, NL database interfaces could finally become a commercial technology. By asking existing database administrators to simply keep logs of the NL queries they receive and the formal (e.g. SQL) queries they construct in response, the requisite corpus of annotated data could be assembled quite easily.

Another way to obtain the requisite supervision is to allow ordinary users themselves to provide the necessary feedback. One approach to allowing a system to learn from its user community after it is deployed is to ask users to confirm a correct interpretation when the system finds a query to be ambiguous. If a system (or an ensemble of several different systems) finds a query ambiguous and produces multiple alternative formal interpretations, the competing queries can be paraphrased back into NL and the user asked to pick the correct one. The chosen interpretation can then be used as a new training example to improve the system. We have recently produced a generation system that produces natural language from formal queries by inverting our WASP system to translate in the opposite direction. Such an NL generation system could be used paraphrase alternative formal interpretations for the user.

Nevertheless, developing training corpora in which each sentence is manually annotated with a detailed formal MR is typically a very difficult and time-consuming process. Ideally, a system would be able to learn language like a human child, by being exposed to utterances in a rich perceptual context. By inferring the meaning of a sentence from the context in which it was uttered, a sentence-meaning pair could be automatically constructed. Methods for inducing semantic parsers from sentences annotated with MRs could then be applied to the resulting data. Although in general it is not possible to infer a unique meaning for a sentence from context, in the vast majority of cases, the context greatly restricts its range of possible meanings. There has been some work on inferring the meanings of individual words given a corpus of sentences each paired with an ambiguous set of multiple possible MRs [33]; however, unlike our work on semantic parsing, this work does not address the issues of learning to disambiguate words and phrases and compose their meanings into semantic representations of complete sentences.

The general problem of *symbol grounding*, how the meaning of abstract symbols is grounded in an agent's perceptual environment and experience, has been argued to be a critical issue in developing truly intelligent artificial systems [19]. Clearly, a deep understanding of most natural language requires capturing the connection between the abstract concepts underlying words and phrases and their embodiment in the physical world. There has been some recent work on inferring a grounded meaning of individual words or short referring expressions from visual perceptual context [31, 2, 42]. However, the syntactic complexity of the natural language used in this work is very restrictive, many of the systems use existing knowledge of the language, and most of them use static images to learn language describing objects and cannot use dynamic video to learn language describing actions. None of this existing work makes use of modern statistical-NLP parsing techniques or learns to build detailed symbolic meaning representations of complete, complex sentences. Developing robust systems that can learn to semantically interpret complex natural language given only exposure to utterances in a perceptual context is a very challenging and important problem for future research. Addressing this problem will require tightly integrating a variety of techniques from computational linguistics, machine learning, knowledge representation, computer vision, and robotics.

6 Conclusions

Semantic parsing is an important task that has a variety of interesting applications. This paper has reviewed several systems that we have developed for learning semantic parsers from corpora annotated with formal meaning representations. Results on automatically acquiring NL interfaces to databases and simulated robotic systems were used to demonstrate the capabilities of our existing systems. There are a number of challenging problems for future research and hopefully this paper will motivate more researchers to explore new methods for automatically acquiring parsers that can produce complete, formal semantic representations.

Acknowledgements

I would like to thank Ruifang Ge, Yuk Wah Wong, and Rohit Kate for developing and evaluating the systems discussed in this paper and John Zelle and Cynthia Thompson for important earlier contributions. This research was supported by the U.S. Defense Advanced Research Projects Agency under grant HR0011-04-1-0007 and a gift from Google Inc. All opinions expressed are solely those of the author.

References

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

2. David Bailey, Jerome Feldman, Srini Narayanan, and George Lakoff. Modeling embodied lexical development. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, 1997.
3. Borland International. *Turbo Prolog 2.0 Reference Guide*. Borland International, Scotts Valley, CA, 1988.
4. Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.
5. Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312, June 1993.
6. Razvan C. Bunescu and Raymond J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, BC, 2005.
7. Razvan C. Bunescu and Raymond J. Mooney. Subsequence kernels for relation extraction. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18: Proceedings of the 2005 Conference*, 2006.
8. Xavier Carreras and Luis Marquez. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, Boston, MA, 2004.
9. 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. Users manual: RoboCup soccer server manual for soccer server version 7.07 and later, 2003. Available at `http://sourceforge.net/projects/sserver/`.
10. David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 263–270, Ann Arbor, MI, June 2005.
11. Michael Collins. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pages 263–270, Philadelphia, PA, July 2002.
12. Michael J. Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 16–23, 1997.
13. Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
14. Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.
15. Jay Earley. An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery*, 6(8):451–455, 1970.
16. Charles J. Fillmore, Baker, Collin F., and Hiroaki Sato. The FrameNet database and software tools. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, pages 1157–1160, Las Palmas, 2002.
17. R. Ge and R. J. Mooney. A statistical semantic parser that integrates syntax and semantics. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 9–16, Ann Arbor, MI, July 2005.

18. R. Ge and R. J. Mooney. Discriminative reranking for semantic parsing. In *Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL-06)*, Sydney, Australia, July 2006.
19. S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 2004.
20. G. G. Hendrix, E. Sacerdoti, D. Sagalowicz, and J. Slocum. Developing a natural language interface to complex data. *ACM Transactions on Database Systems*, 3(2):105–147, 1978.
21. E. H. Hovy, M. Marcus, M. Palmer, S. Pradhan, L. Ramshaw, and R. Weischedel. OntoNotes: The 90% solution. In *Proceedings of Human Language Technology Conference / North American Association for Computational Linguistics Annual Meeting (HLT-NAACL-06)*, New York, NY, 2006.
22. Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the Tenth European Conference on Machine Learning (ECML-98)*, pages 137–142, Berlin, 1998. Springer-Verlag.
23. R. Kate and R. J. Mooney. Using string-kernels for learning semantic parsers. In *Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL-06)*, Sydney, Australia, July 2006.
24. R. J. Kate, Y. W. Wong, and R. J. Mooney. Learning to transform natural to formal languages. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-2005)*, pages 1062–1068, Pittsburgh, PA, July 2005.
25. G. Kuhlmann, P. Stone, R.J. Mooney, and J.W. Shavlik. Guiding a reinforcement learner with natural language advice: Initial results in RoboCup soccer. In *Proceedings of the AAAI-04 Workshop on Supervisory Control of Learning and Adaptive Systems*, San Jose, CA, July 2004.
26. Nada Lavrac and Saso Dzeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, 1994.
27. Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.
28. Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.
29. Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates. Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability. In *Proceedings of the Twentieth International Conference on Computational Linguistics (COLING-04)*, Geneva, Switzerland, August 2004.
30. Stefan Riezler, Detlef Prescher, Jonas Kuhn, and Mark Johnson. Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, pages 480–487, Hong Kong, 2000.
31. Deb Roy. Learning visually grounded words and syntax for a scene description task. *Computer Speech and Language*, 16(3):353–385, 2002.
32. John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
33. Jeffrey M. Siskind. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61(1):39–91, October 1996.
34. Andreas Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201, 1995.

35. L. R. Tang and R. J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Proceedings of the 12th European Conference on Machine Learning*, pages 466–477, Freiburg, Germany, 2001.
36. Lappoon R. Tang and Raymond J. Mooney. Automated construction of database interfaces: Integrating statistical and relational learning for semantic parsing. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora(EMNLP/VLC-2000)*, pages 133–141, Hong Kong, October 2000.
37. Cynthia A. Thompson and Raymond J. Mooney. Acquiring word-meaning mappings for natural language interfaces. *Journal of Artificial Intelligence Research*, 18:1–44, 2003.
38. David L. Waltz. An English language question answering system for a large relational database. *Communications of the Association for Computing Machinery*, 21(7):526–539, 1978.
39. Yuk Wah Wong and Raymond J. Mooney. Learning for semantic parsing with statistical machine translation. In *Proceedings of the 2006 Human Language Technology Conference - North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL-06)*, pages 439–446, New York City, NY, June 2006.
40. William A. Woods. Lunar rocks in natural English: Explorations in natural language question answering. In Antonio Zampoli, editor, *Linguistic Structures Processing*. Elsevier North-Holland, New York, 1977.
41. Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, pages 523–530, Toulouse, France, 2001.
42. Chen Yu and Dana H. Ballard. On the integration of grounding language and learning objects. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-2004)*, pages 488–493, 2004.
43. D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106, 2003.
44. John M. Zelle and Raymond J. Mooney. 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 1996.
45. Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of 21th Conference on Uncertainty in Artificial Intelligence (UAI-2005)*, Edinburgh, Scotland, July 2005.
46. Victor W. Zue and James R. Glass. Conversational interfaces: Advances and challenges. In *Proceedings of the IEEE*, volume 88(8), pages 1166–1180, 2000.