Semi-Supervised Learning for Semantic Parsing using Support Vector Machines

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

Abstract

We present a method for utilizing unannotated sentences to improve a semantic parser which maps natural language (NL) sentences into their formal meaning representations (MRs). Given NL sentences annotated with their MRs, the initial supervised semantic parser learns the mapping by training Support Vector Machine (SVM) classifiers for every production in the MR grammar. Our new method applies the learned semantic parser to the unannotated sentences and collects unlabeled examples which are then used to retrain the classifiers using a variant of *transductive* SVMs. Experimental results show the improvements obtained over the purely supervised parser, particularly when the annotated training set is small.

1 Introduction

Semantic parsing is the task of mapping a natural language (NL) sentence into a complete, formal *meaning representation* (MR) which a computer program can execute to perform some task, like answering database queries or controlling a robot. These MRs are expressed in domain-specific unambiguous formal *meaning representation languages* (MRLs). Given a training corpus of NL sentences annotated with their correct MRs, the goal of a learning system for semantic parsing is to induce an efficient and accurate semantic parser that can map novel sentences into their correct MRs.

Several learning systems have been developed for semantic parsing, many of them recently (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Ge and Mooney, 2005; Kate and Mooney, 2006). These systems use supervised learning methods which only utilize annotated NL sentences. However, it requires considerable human effort to annotate sentences. In contrast, unannotated NL sentences are usually easily available. Semi-supervised learning methods utilize cheaply available unannotated data during training along with annotated data and often perform better than purely supervised learning methods trained on the same amount of annotated data (Chapelle et al., 2006). In this paper we present, to our knowledge, the first semi-supervised learning system for semantic parsing.

We modify KRISP, a supervised learning system for semantic parsing presented in (Kate and Mooney, 2006), to make a semi-supervised system we call SEMISUP-KRISP. Experiments on a real-world dataset show the improvements SEMISUP-KRISP obtains over KRISP by utilizing unannotated sentences.

2 Background

This section briefly provides background needed for describing our approach to semi-supervised semantic parsing.

2.1 KRISP: The Supervised Semantic Parsing Learning System

KRISP (Kernel-based Robust Interpretation for Semantic Parsing) (Kate and Mooney, 2006) is a supervised learning system for semantic parsing which

Proceedings of NAACL HLT 2007, Companion Volume, pages 81–84,
Rochester, NY, April 2007. ©2007 Association for Computational Linguistics

takes NL sentences paired with their MRs as training data. The productions of the formal MRL grammar are treated like semantic concepts. For each of these productions, a Support-Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2000) classifier is trained using string similarity as the kernel (Lodhi et al., 2002). Each classifier can then estimate the probability of any NL substring representing the semantic concept for its production. During semantic parsing, the classifiers are called to estimate probabilities on different substrings of the sentence to compositionally build the most probable meaning representation (MR) of the sentence.

KRISP trains the classifiers used in semantic parsing iteratively. 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 corresponding MRs use the production π in their parse trees. The set of negative examples includes all of the other training sentences. Using these positive and negative examples, an SVM classifier is trained for each production π using a string kernel. In subsequent iterations, the parser learned from the previous iteration is applied to the training examples and more refined positive and negative examples, which are more specific substrings within the sentences, are collected for training. Iterations are continued until the classifiers converge, analogous to iterations in EM (Dempster et al., 1977). Experimentally, KRISP compares favorably to other existing semantic parsing systems and is particularly robust to noisy training data (Kate and Mooney, 2006).

2.2 Transductive SVMs

SVMs (Cristianini and Shawe-Taylor, 2000) are state-of-the-art machine learning methods for classification. Given positive and negative training examples in some vector space, an SVM finds the maximum-margin hyperplane which separates them. Maximizing the margin prevents over-fitting in very high-dimensional data which is typical in natural language processing and thus leads to better generalization performance on test examples. When the unlabeled test examples are also available during training, a transductive framework for learning (Vapnik, 1998) can further improve the performance on the test examples.

Transductive SVMs were introduced in (Joachims, 1999). The key idea is to find the labeling of the test examples that results in the maximum-margin hyperplane that separates the positive and negative examples of *both* the training and the test data. This is achieved by including variables in the SVM's objective function representing labels of the test examples. Finding the exact solution to the resulting optimization problem is intractable, however Joachims (1999) gives an approximation algorithm for it. One drawback of his algorithm is that it requires the proportion of positive and negative examples in the test data be close to the proportion in the training data, which may not always hold, particularly when the training data is small. Chen et al. (2003) present another approximation algorithm which we use in our system because it does not require this assumption. More recently, new optimization methods have been used to scale-up transductive SVMs to large data sets (Collobert et al., 2006), however we did not face scaling problems in our current experiments.

Although transductive SVMs were originally designed to improve performance on the *test* data by utilizing its availability during training, they can also be directly used in a semi-supervised setting (Bennett and Demiriz, 1999) where unlabeled data is available during training that comes from the same distribution as the test data but is not the actual data on which the classifier is eventually to be tested. This framework is more realistic in the context of semantic parsing where sentences must be processed in real-time and it is not practical to re-train the parser transductively for every new test sentence. Instead of using an alternative semi-supervised SVM algorithm, we preferred to use a transductive SVM algorithm (Chen et al., 2003) in a semi-supervised manner, since it is easily implemented on top of an existing SVM system.

3 Semi-Supervised Semantic Parsing

We modified the existing supervised system KRISP, described in section 2.1, to incorporate semi-supervised learning. Supervised learning in KRISP involves training SVM classifiers on positive and negative examples that are substrings of the anno-

```
function TRAIN_SEMISUP_KRISP(Annotated corpus $\mathcal{A} = \{(s_i, m_i)|i = 1..N\}$, MRL grammar $G$,
                             Unannotated sentences $\mathcal{T} = \{t_i|i = 1..M\}$)
$\mathcal{C} \equiv \{C_\pi|\pi \in G\}$ = TRAIN_KRISP(A,G) // classifiers obtained by training KRISP
Let
  $\mathcal{P} = \{p_\pi$ = Set of positive examples used in training $C_\pi|\pi \in G\}$
  $\mathcal{N} = \{n_\pi$ = Set of negative examples used in training $C_\pi|\pi \in G\}$
  $\mathcal{U} = \{u_\pi = \phi|\pi \in G\}$ // set of unlabeled examples for each production, initially all empty
for $i = 1$ to $M$ do
  $\{u_\pi^i|\pi \in G\}$ =COLLECT_CLASSIFIER_CALLS(PARSE($t_i, \mathcal{C}$))
  $\mathcal{U} = \{u_\pi = u_\pi \cup u_\pi^i|\pi \in G\}$
for each $\pi \in G$ do
  $C_\pi$ =TRANSDUCTIVE_SVM_TRAIN($p_\pi, n_\pi, u_\pi$) // retrain classifiers utilizing unlabeled examples
return classifiers $\mathcal{C} = \{C_\pi|\pi \in G\}$
```

Figure 1: SEMISUP-KRISP's training algorithm

tated sentences. In order to perform semi-supervised learning, these classifiers need to be given appropriate unlabeled examples. The key question is: Which substrings of the unannotated sentences should be given as unlabeled examples to which productions' classifiers? Giving all substrings of the unannotated sentences as unlabeled examples to all of the classifiers would lead to a huge number of unlabeled examples that would not conform to the underlying distribution of classes each classifier is trying to separate. SEMISUP-KRISP's training algorithm, described below and shown in Figure 1, addresses this issue.

The training algorithm first runs KRISP's existing training algorithm and obtains SVM classifiers for every production in the MRL grammar. Sets of positive and negative examples that were used for training the classifiers in the last iteration are collected for each production. Next, the learned parser is applied to the unannotated sentences. During the parsing of each sentence, whenever a classifier is called to estimate the probability of a substring representing the semantic concept for its production, that substring is saved as an unlabeled example for that classifier. These substrings are representative of the examples that the classifier will actually need to handle during testing. Note that the MRs obtained from parsing the unannotated sentences do not play a role during training since it is unknown whether or not they are correct. These sets of unlabeled examples for each production, along with the sets of positive and negative examples collected earlier, are then used to retrain the classifiers using transductive SVMs. The retrained classifiers are finally returned and used in the final semantic parser.

4 Experiments

We compared the performance of SEMISUP-KRISP and KRISP in the GEOQUERY domain for semantic parsing in which the MRL is a functional language used to query a U.S. geography database (Kate et al., 2005). This domain has been used in most of the previous work. The original corpus contains 250 NL queries collected from undergraduate students and annotated with their correct MRs (Zelle and Mooney, 1996). Later, 630 additional NL queries were collected from real users of a web-based interface and annotated (Tang and Mooney, 2001). We used this data as *unannotated* sentences in our current experiments. We also collected an additional 407 queries from the same interface, making a total of $1,037$ unannotated sentences.

The systems were evaluated using standard 10-fold cross validation. All the unannotated sentences were used for training in each fold. Performance was measured in terms of precision (the percentage of generated MRs that were correct) and recall (the percentage of all sentences for which correct MRs were obtained). An output MR is considered correct if and only if the resulting query retrieves the same answer as the correct MR when submitted to the database. Since the systems assign confidences to the MRs they generate, the entire range of the precision-recall trade-off can be obtained for a system by measuring precision and recall at various confidence levels. We present learning curves for the best F-measure (harmonic mean of precision and re-

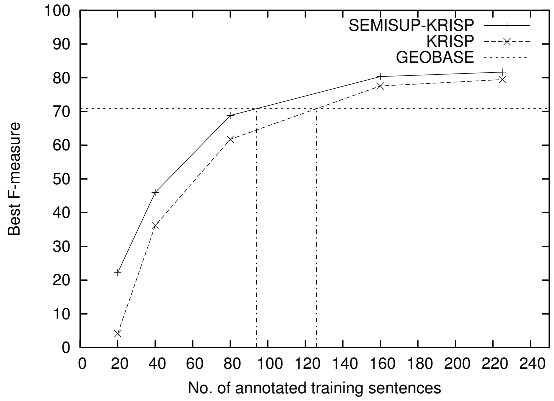

Figure 2: Learning curves for the best F-measures on the GEOQUERY corpus.

call) obtained across the precision-recall trade-off as the amount of annotated training data is increased. Figure 2 shows the results for both systems.

The results clearly show the improvement SEMISUP-KRISP obtains over KRISP by utilizing unannotated sentences, particularly when the number of annotated sentences is small. We also show the performance of a hand-built semantic parser GEOBASE (Borland International, 1988) for comparison. From the figure, it can be seen that, on average, KRISP achieves the same performance as GEOBASE when it is given 126 annotated examples, while SEMISUP-KRISP reaches this level given only 94 annotated examples, a 25.4% savings in human-annotation effort.

5 Conclusions

This paper has presented a semi-supervised approach to semantic parsing. Our method utilizes unannotated sentences during training by extracting unlabeled examples for the SVM classifiers it uses to perform semantic parsing. These classifiers are then retrained using transductive SVMs. Experimental results demonstrated that this exploitation of unlabeled data significantly improved the accuracy of the resulting parsers when only limited supervised data was provided.

Acknowledgments

This research was supported by a Google research grant. The experiments were run on the Mastodon cluster provided by NSF grant EIA-0303609.

References

K. Bennett and A. Demiriz. 1999. Semi-supervised support vector machines. *Advances in Neural Information Processing Systems*, 11:368–374.

Borland International. 1988. *Turbo Prolog 2.0 Reference Guide*. Borland International, Scotts Valley, CA.

O. Chapelle, B. Schölkopf, and A. Zien, editors. 2006. *Semi-Supervised Learning*. MIT Press, Cambridge, MA.

Y. Chen, G. Wang, and S. Dong. 2003. Learning with progressive transductive support vector machine. *Pattern Recognition Letters*, 24:1845–1855.

R. Collobert, F. Sinz, J. Weston, and L. Bottou. 2006. Large scale transductive SVMs. *Journal of Machine Learning Research*, 7(Aug):1687–1712.

N. Cristianini and J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38.

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.

T. Joachims. 1999. Transductive inference for text classification using support vector machines. In *Proc. of ICML-99*, pages 200–209, Bled, Slovenia, June.

R. J. Kate and R. J. Mooney. 2006. Using string-kernels for learning semantic parsers. In *Proc. of COLING/ACL-06*, pages 913–920, Sydney, Australia, 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.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.

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.

V. N. Vapnik. 1998. *Statistical Learning Theory*. John Wiley & Sons.

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.