In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07)*. pp. 895-900, Vancouver, Canada, July 2007.

Learning Language Semantics from Ambiguous Supervision

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

This paper presents a method for learning a semantic parser from ambiguous supervision. Training data consists of natural language sentences annotated with multiple potential meaning representations, only one of which is correct. Such ambiguous supervision models the type of supervision that can be more naturally available to language-learning systems. Given such weak supervision, our approach produces a semantic parser that maps sentences into meaning representations. An existing semantic parsing learning system that can only learn from unambiguous supervision is augmented to handle ambiguous supervision. Experimental results show that the resulting system is able to cope up with ambiguities and learn accurate semantic parsers.

Introduction

Most learning systems for natural-language processing require very detailed supervision in the form of human annotation such as parse trees or semantic labels. Ideally, a system would be able to learn language like a human child, by only being exposed to utterances in a rich perceptual context. Frequently, the context of an utterance can be used to narrow down its interpretation to a fairly small set of reasonable alternatives. 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 meaning representations (Siskind 1996). However, the methods developed in this work only acquire lexical semantics and do not learn how to disambiguate words and compose their meanings in order to interpret complete sentences.

Recent research in semantic parsing has developed methods that learn to map sentences into complete formal *meaning representations* (MRs) by training on a corpus of sentences each annotated with its corresponding MR (Ge & Mooney 2005; Zettlemoyer & Collins 2005; Kate, Wong, & Mooney 2005). However, labeling each sentence in a corpus with a detailed, correct semantic representation is a difficult task and is also not a realistic model of how children learn to analyze the meaning of sentences. In this paper, we explore the task of learning a semantic parser from ambiguous supervision, in which each sentence is only annotated with an ambiguous set of multiple, alternative potential interpretations. We show how an accurate semantic parser can be learned by augmenting an existing supervised learning system to handle such ambiguous training data. Specifically, we add an iterative retraining method to KRISP (Kate & Mooney 2006), a system that learns a semantic parser using a support-vector machine (SVM) that utilizes a string kernel (Lodhi *et al.* 2002). In a manner analogous to the Expectation Maximization (EM) algorithm (Dempster, Laird, & Rubin 1977), our iterative method is able to converge on accurate meanings for each sentence in an ambiguous corpus by finding stable, reliable patterns in the equivocal training data.

Testing such a system in a realistic setting would require a perceptual system that can construct a set of plausible meanings for a sentence from the context in which it is uttered. Since this is a difficult unsolved problem, we evaluate the system by artificially obfuscating training data previously used to assess supervised semantic-parser learners. By adding additional incorrect meanings to the correct meaning for each sentence, an ambiguous training corpus is created. We also evaluate our system on another artificially-created corpus that models ambiguities more realistically. Experimental results indicate that our system is able to learn accurate parsers even given such ambiguous supervision.

Background

This section provides background on the task of semantic parsing when the training data is unambiguous and briefly describes a learning system designed for this task. The next section describes the extensions we made to this system to handle the case when the training data is ambiguous.

Semantic Parsing with Unambiguous Supervision

Semantic parsing is the task of mapping a natural language (NL) sentence into a computer-executable complete meaning representation (MR). These MRs are expressed in domain-specific *meaning representation languages* (MRL). We call the type of supervision *unambiguous* when a learning system for semantic parsing is given a corpus of NL sentences in which each sentence is paired with its respective correct MR. In this type of supervision, while the learning system is not given which parts of the MRs correspond to

SENT1: *"Which rivers run through the states bordering Texas?"*
MR1: `answer(traverse(next_to(stateid('texas'))))`

SENT2: *"Which is the highest point in Alaska?"*
MR2: `answer(highest(place(loc(stateid('Alaska')))))`

SENT3: *"What are the major rivers in Alaska?"*
MR3: `answer(major(river(loc(stateid('Alaska')))))`

Figure 1: Examples of NL sentences unambiguously paired with their respective correct MRs.

which portions within the sentences, it is however unambiguously given which complete MRs correspond to which sentences.

Figure 1 shows examples of NL sentences unambiguously paired with their respective correct MRs from the GEOQUERY domain in which the MRL is a functional language used to query a database of U.S. geographical facts (Kate, Wong, & Mooney 2005).

KRISP: The Semantic Parsing Learning System for Unambiguous Supervision

KRISP (Kernel-based Robust Interpretation for Semantic Parsing; Kate & Mooney 2006) is a system that learns a semantic parser from unambiguous training data. It treats the productions of the formal MRL grammar as semantic concepts. For each of these productions, a Support-Vector Machine (SVM; Cristianini & Shawe-Taylor 2000) classifier is trained using string similarity as the kernel (Lodhi *et al.* 2002). Each classifier then estimates the probability of different NL substrings representing the semantic concept for its production. During semantic parsing, the classifiers are called to estimate probabilities on different substrings of the input sentence to compositionally build the most probable MR for the complete sentence. The probability assigned to an MR represents the confidence the parser has in its output. Given an MR and a sentence, KRISP can also compute a confidence that this MR is the correct one for the given sentence. This function, which we call PARSE_ESTIMATE, is also used internally in KRISP's training algorithm.

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 sentences and the PARSE_ESTIMATE function is applied to their correct MRs, 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. Experimentally, KRISP compares favorably to other existing semantic parsing systems and is particularly robust to noisy training data (Kate & Mooney 2006).

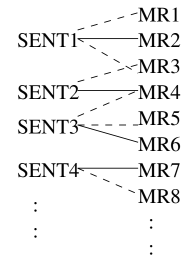

Figure 2: Sample Ambiguous Training Data (solid-line: correct meaning; dashed-line: possible meaning)

Learning Semantic Parsers from Ambiguous Supervision

There are two major shortcomings with the type of detailed supervision described in the previous section. First, manually constructing an unambiguous corpus in which each sentence is annotated with its correct MR is a difficult task. A computer system that observes some perceptual context and is simultaneously exposed to natural language should be able to automatically learn the underlying language semantics. But since the training data available in such a setting will not consist of NL sentences unambiguously paired with their MRs, it will require human effort to build such a corpus before the learning can take place. Secondly, unambiguous supervision does not model the type of data children receive when they are learning a language. In order to learn to analyze the meaning of sentences, children have to also learn to identify the correct meaning of a sentence among the several meanings possible in their current perceptual context. Therefore, a weaker and more general form of supervision for learning semantic parsers needs to be considered.

Ambiguous Supervision

Consider the type of supervision a system would receive when learning language semantics from perceptual contexts. We assume that the low-level sensory data (real or simulated) from a system's perceptual context is first abstracted into symbolic meaning representations (MRs). Our model of supervision corresponds to the type of data that will be gathered from a temporal sequence of perceptual contexts with occasional language commentary. The training data in this model thus consists of a sequence of NL sentences and a sequence of MRs in which each NL sentence is associated with a set of one or more consecutive MRs. These associated MRs represent the general perceptual context the system was in when it registered the NL sentence. The association of multiple MRs with each NL sentence makes such a corpus ambiguous.

We assume that each NL utterance means something unique in the perceptual context, so that *exactly one* MR out of all the MRs associated with an NL sentence represents its correct meaning. Also, since different NL utterances would normally refer to different things, we assume that an MR can be the correct meaning of *at most one* of the sentences with which it is associated.[1] Figure 2 shows a small exam-

[1]A duplicate MR (or sentence) that reappears later in the sequence will be treated separately each time.

function TRAIN_KRISPER(Ambiguous corpus $\mathcal{A} = \{(sent_i, \mathcal{M}_i) | \mathcal{M}_i \neq \phi, i = 1..N\}$, MRL grammar G)
REDUCE_AMBIGUITIES($\mathcal{A}$) // reduce easy to resolve ambiguities
$\mathcal{U} = \phi$ // corpus for training, initially empty
for $i = 1$ **to** N **do** // collect weighted initial examples
  $\mathcal{U} = \mathcal{U} \cup \{(sent_i, mr, w) | mr \in \mathcal{M}_i, w = 1/|\mathcal{M}_i|\}$
$\mathcal{C} \equiv \{C_\pi | \pi \in G\}$ = TRAIN_WEIGHTED_KRISP($\mathcal{U}$,G) // classifiers obtained by training KRISP
while (not converged) **do**
  $\mathcal{V} = \phi$ // collect examples with parse confidences, initially empty
  for $i = 1$ **to** N **do**
    $\mathcal{V} = \mathcal{V} \cup \{(sent_i, mr, w) | mr \in \mathcal{M}_i, w =$PARSE_ESTIMATE$(sent_i, mr, \mathcal{C})\}$
  $\mathcal{U}$ =BEST_EXAMPLES($\mathcal{V}$) // find the best consistent examples
  $\mathcal{C}$ =TRAIN_KRISP($\mathcal{U}$,G) // retraining
return $\mathcal{C}$ // return classifiers trained in the last iteration

Figure 3: KRISPER's training algorithm

ple of such a corpus. The sentences are shown connected to their possible MRs by lines. For illustration purpose, the connections between sentences and their correct MRs are shown with solid lines and the rest are shown with dotted lines. However, this distinction is obviously not included in the training data.

KRISPER: The Semantic Parsing Learning System for Ambiguous Supervision

We extended KRISP's training algorithm to handle ambiguous supervision. We call our new system KRISPER (KRISP with EM-like Retraining). It employs an iterative approach analogous to EM, where, through retraining, each iteration improves upon determining the correct MR out of the possible MRs for each sentence.

Figure 3 shows KRISPER's training algorithm. It takes the ambiguous corpus as input in which each NL sentence is paired with a non-empty set of MRs. Using the assumption that an MR in the corpus can be the correct meaning of at most one sentence, it first removes some easily resolved ambiguities present in the input. For example, if the input includes the examples $(sent_1, \{mr_1\})$, $(sent_2, \{mr_1, mr_2\})$ and $(sent_3, \{mr_2, mr_3, mr_4\})$, then it is clear that mr_1 can not be the correct meaning of $sent_2$, because then $sent_1$ will be left without any correct meaning, hence mr_2 must be the correct meaning of $sent_2$. This then prohibits mr_2 from being the correct meaning of $sent_3$, reducing the set of its possible MRs to $\{mr_3, mr_4\}$. In general, we use the following procedure to remove these type of ambiguities. First, we note that an ambiguous corpus forms a bipartite graph with the sentences and the MRs as two disjoint sets of vertices and the associations between them as connecting edges. The set of correct NL–MR pairs form a *matching* on this bipartite graph which is defined as a subset of the edges with at most one edge incident on every vertex (Cormen, Leiserson, & Rivest 1990). Since all NL sentences have a correct MR, this matching is in fact a *maximum matching* with cardinality equal to the number of sentences. In order to check whether an NL–MR pair can be in the set of the correct NL–MR pairs, our procedure removes the edge connecting the pair and the edges incident on the two vertices and sees if the maximum matching on the resulting graph includes all of the remaining NL sentences.[2] If not, then the procedure removes that NL–MR association from the corpus, because any matching that includes it will not be able to include edges to cover all of the NL sentences. All of the NL–MR pairs in the corpus are checked by this procedure, and the whole process is iterated until no additional NL–MR pairs are removed.

KRISPER then assumes that *every* MR in the set of MRs paired with each NL sentence is correct for that NL sentence and collects the resulting NL–MR paired examples. Each such example is also given a weight which is inversely proportional to the number of MRs that were associated with its NL sentence. For example, from the ambiguous input example of $(sent_3, \{mr_3, mr_4\})$, two examples, $(sent_3, mr_3)$ and $(sent_3, mr_4)$, will be collected and both will be given weight equal to $1/2$. These examples are then given to KRISP's training algorithm (described in the "Background" section) to learn an initial semantic parser. Since the training data is noisy with many incorrect NL–MR pairs in this first iteration, the parameter of the SVM training algorithm that penalizes incorrect classifications is kept low. This parameter is later increased with each subsequent iteration. Although there will be many incorrect NL–MR pairs in the first iteration, the parser is expected to learn some regularities due to the presence of the correct pairs. The weighting procedure ensures that the more an NL–MR pair is likely to be incorrect, the less weight it receives. KRISP's existing training algorithm, however, does not accept weighted examples as input so we modified it as follows. All of the positive and negative SVM training examples that KRISP extracts from an input NL–MR pair are given the same weight as the input example. Then, a version of an SVM which takes weighted input examples is used for training. We used the tool "Weights for data instances" available in the LIBSVM package.[3]

Next, for each NL sentence in the training data, KRISPER estimates the confidence of generating each of the MRs in

[2] An $O(|V||E|)$ algorithm exists for finding a maximum matching on a bipartite graph, where $|V|$ is the number of vertices and $|E|$ is the number of edges (Cormen, Leiserson, & Rivest 1990). Our procedure finds a maximum matching separately for every maximally connected subgraph of the graph which makes this step even more efficient.

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools

the sentence's set of possible MRs by calling the learned parser's PARSE_ESTIMATE function. For the purpose of training the parser in the next and subsequent iterations, it pairs the NL sentence with only one MR from its set, the one with the highest confidence.[4] But since the sets of possible MRs of NL sentences could overlap, the new NL–MR pairs should be consistently chosen so that an MR does not get paired with more than one sentence. The problem of consistently selecting the best pairs is an instance of the *maximum weight assignment* problem on a bipartite graph which can be solved using the *Hungarian Algorithm* (Munkres 1957) in $O(|V|^3)$ time, where $|V|$ is the number of vertices. The pairs found by this algorithm are then given to KRISP's training algorithm to learn a better semantic parser. Since now only one MR is associated with each NL sentence, the weights of all examples are set to 1. The algorithm terminates when the NL–MR pairs for an iteration differ by less than 5% (a parameter) from the pairs in the previous iteration. The parser trained on these NL–MR pairs is then returned as the final learned parser.

Experiments

This section presents an experimental evaluation of KRISPER. To our best knowledge, there is no real-world ambiguous corpus available for learning semantic parsers. Hence, to evaluate our system, we constructed two ambiguous corpora: AMBIG-GEOQUERY and AMBIG-CHILDWORLD. The AMBIG-GEOQUERY corpus was constructed by artificially obfuscating the existing real-world unambiguous GEOQUERY corpus (described in the "Background" section), while the AMBIG-CHILDWORLD corpus was constructed completely artificially but attempts to more accurately model real-world ambiguities. The following subsection describes how the two corpora were constructed. The next subsections describe the experimental evaluation of KRISPER's performance on both corpora with different levels of ambiguity.

Corpora Construction

Ambig-Geoquery Corpus

We used the the unambiguous GEOQUERY corpus to artificially construct the AMBIG-GEOQUERY corpus which conforms to the model of ambiguous supervision described earlier with a sample shown in figure 2. Using the MRs and NL sentences present in the unambiguous corpus, an ambiguous corpus was formed in which each NL sentence was paired with a set of multiple possible MRs. First, a sequence of randomly permuted MRs from the GEOQUERY corpus, which we call *base* MRs, was formed. Next, random MRs, chosen from the same corpus, were inserted between every pair of adjacent base MRs. The number of MRs inserted between any two base MRs was randomly chosen uniformly between 0 and α (a parameter). Next, each NL sentence

[4] We found that this works better than pairing each NL sentence with all of its associated MRs with weights proportional to their confidences because after the first iteration the system usually has a good idea of the correct NL–MR pairs and the remaining pairs only increase the noise.

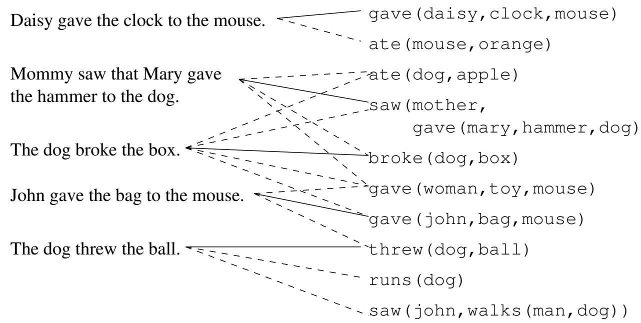

Figure 4: A sample of the AMBIG-CHILDWORLD corpus corresponding to a perceptual context.

from the GEOQUERY corpus was paired with a set of MRs that formed a window in the sequence centered at the sentence's correct base MR. The width of the window in either direction from the base MR was randomly chosen uniformly between 0 and β (another parameter). Since the window is around the correct MR, this ensures that there will always be a correct meaning for each sentence in its set of possible MRs.

By varying the parameters α and β, we generated three levels of ambiguity, which we call levels 1, 2 and 3. In level 1, both parameters were set to 1 and this resulted in training data that on average has 24.8% sentences associated with only one MR, 50.1% with two and 25.1% with three MRs. In level 2, both parameters were set to 2 and this resulted in an average of 11.2% sentences associated with one MR, 22.3% with two, 33.7% with three, 22% with four and 10.8% with five MRs. Finally, in level 3, both parameters were set to 3 and there were on average 6% sentences associated with only one MR, 12.5% with two, 19.4% with three, 25.9% with four, 18.4% with five, 11.8% with six and 6% with seven MRs.

Ambig-ChildWorld Corpus

Although the AMBIG-GEOQUERY corpus uses real-world NL sentences and MRs, it does not model the ambiguities realistically because the MRs associated with a sentence may have nothing in common, but in a real-world perceptual context, the potential candidate meanings for a sentence will usually be related. Hence, we created another corpus, AMBIG-CHILDWORLD, which models ambiguities more realistically. It tries to mimic the type of language data that would be available to a child while learning a language.

We first constructed a synchronous context-free grammar (Aho & Ullman 1972) to simultaneously generate simple NL sentences and their correct MRs which are in predicate logic but without quantification. The synchronous grammar relates some simple NL verbs to logic predicates and some simple NL nouns (namely people, animals and things) to the arguments of those predicates. There are 15 verbs and 37 nouns in the grammar. It can also generate a few complex sentences. The corpus is generated to model occasional language commentary on a series of perceptual contexts. A perceptual context is modelled like a collection of events happening in a room involving a few people, animals and things. The next perceptual context will involve different people, animals and things.

The data corresponding to a perceptual context was created in the following manner. First, a random subset of the synchronous grammar was extracted. Since people, animals and things are part of the grammar, this process selects random subsets of them as well. Next, a random sequence of NL sentences and their correct MRs was generated using this subset of the synchronous grammar, but only a few of the NL sentences were retained. These were chosen in such a way that the number of skipped sentences between a retained sentence and the next retained sentence in the sequence was randomly chosen uniformly between 0 and α, where α is a parameter.[5] Retaining only a few sentences models the fact that only a few events happening in the room will receive NL commentary. Next, each NL sentence was associated with a set of MRs which form a window in the sequence centered at the sentence's correct MR. The width of the window in either direction of the MR was again randomly chosen uniformly between 0 and β. This models the fact that the sentence might be referring to one of the multiple events that were happening while the sentence was being spoken.

The process of generating data then continues with a different subset of the grammar representing a different perceptual context. Since the MRs generated from a perceptual context will involve the same people, animals and things, they will usually be related, thus modelling ambiguities more realistically. From each perceptual context, 5 to 10 NL sentences with their associated MR sets were created. A sample of the AMBIG-CHILDWORLD corpus corresponding to one perceptual context is shown in figure 4. For illustration, the correct NL–MR pairs are shown with solid lines and the remaining pairs with dotted lines. In this corpus there were on average 5.47 words in a sentence.

Three levels of ambiguity were created for this corpus by varying the parameters α and β similar to the way in which the three levels of ambiguity for the AMBIG-GEOQUERY corpus were created. The distributions of the number of MRs associated with NL sentences in the three levels of this corpus are also similar to the distributions in the corresponding levels of the AMBIG-GEOQUERY corpus. On the two corpora, the first step of KRISPER's training algorithm, which removes easily resolvable ambiguities, removed on average 8% NL–MR pairs for level 1, 3.2% pairs for level 2 and 1.3% pairs for level 3.

Methodology

KRISPER was evaluated using standard 10-fold cross validation. For the AMBIG-GEOQUERY corpus, the original unambiguous GEOQUERY corpus was divided into ten equal parts. For each fold, one part was used for testing the accuracy of the learned semantic parser and the remaining nine parts were used to construct the ambiguous training data by the method described in the previous subsection. Since AMBIG-CHILDWORLD is artificially created, there was no scarcity of training data, hence the training data for each fold was generated separately. The testing data for each fold

[5] It plays the same role here as in the construction of the AMBIG-GEOQUERY corpus.

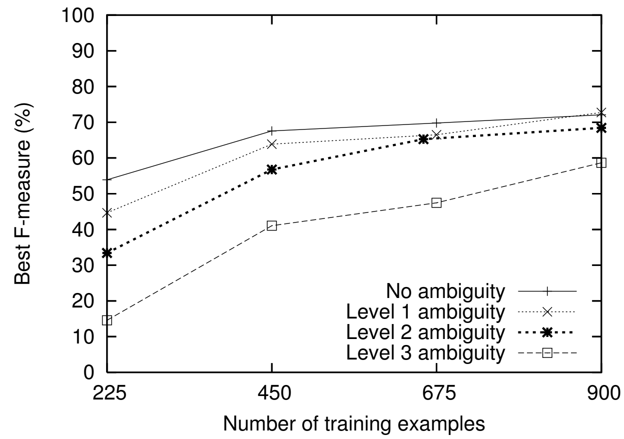

Figure 5: Learning curves for KRISPER on the AMBIG-GEOQUERY corpus with various levels of ambiguities.

was also generated separately using the entire synchronous grammar, but with no ambiguity added.

There are some constants in the GEOQUERY domain, like state and city names, which appear in the NL sentences as well as in their corresponding MRs. This information is normally exploited when training KRISP, but if it is exploited in the obfuscated data then the introduced ambiguities can often be trivially resolved because most of the time only one MR out of the possible candidate MRs will have the constants which are present in the associated sentence. Therefore, we prevented KRISPER from exploiting such matching constants, making the learning task even harder because the parser now has to learn the meanings of the constants as well (which is analogous to learning the names of specific items in the world). This is also true with the AMBIG-CHILDWORLD corpus.

The unambiguous GEOQUERY corpus contains 250 NL queries annotated with their correct MRs (Zelle & Mooney 1996). Since learning accurate parsers from ambiguous data obviously requires more examples than when using unambiguous data, we also tried artificially increasing the size of the AMBIG-GEOQUERY training set by replicating examples in the training set after changing their constants to other constants of the same type (e.g. changing a state name to another state name). The other constants were always chosen from within the training corpus being replicated. In our experiments, the corpus was replicated two, three and four times for each of the three levels of ambiguities. We also created the training data for the AMBIG-CHILDWORLD corpus for each ambiguity level in the same four sizes: each fold containing 225, 450, 675 and 900 examples. The testing corpus was same for all the sizes, each fold containing 25 examples.

For higher levels of ambiguity when the training size is large, the number of NL–MR pairs generated in the first iteration of KRISPER's training algorithm becomes very large. To save running time, we subsample the training data in the first iteration to result in size of around 500 NL–MR pairs per sample. The subsampling is not done in the subsequent iterations though. In our experiments, KRISPER's training

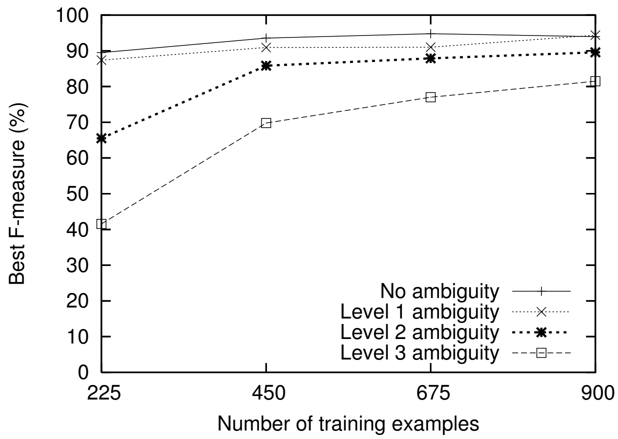

Figure 6: Learning curves for KRISPER on the AMBIG-CHILDWORLD corpus with various levels of ambiguities.

algorithm did not require more than six iterations to converge.

Performance of semantic parsing 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). For AMBIG-GEOQUERY corpus, an output MR is considered correct if the resulting query retrieves the same answer as the correct MR when submitted to the database, and for AMBIG-CHILDWORLD corpus, an output MR is considered correct if it exactly matches the correct MR.

Since KRISPER assigns confidences to the MRs it outputs, an entire range of the precision-recall trade-off can be obtained by varying the confidence threshold for accepting a parse. We present the results in the form of learning curves for the best F-measure (harmonic mean of precision and recall) across the precision-recall tradeoff.

Results and Discussion

Figure 5 shows the results obtained by training KRISPER on the AMBIG-GEOQUERY training data with ambiguity levels 1, 2 and 3. The results obtained when the training data is unambiguous are also shown for comparison. When smaller number of training examples are given, the performance on ambiguous training data for all the levels is worse than the performance on unambiguous training data. But as the amount of ambiguous training data is increased, despite the weak form of supervision, KRISPER starts to learn as accurate a parser as with the same amount of unambiguous training data. The learning curve for level 3 shows a large performance gain but it has not yet converged.

Figure 6 shows the results obtained when training KRISPER on the AMBIG-CHILDWORLD data with the three ambiguity levels. The results for unambiguous training are also shown for comparison. As the training-set size is increased, KRISPER is able to overcome the ambiguities and learn almost as accurate a semantic parser as with no ambiguity on this corpus as well. Since weaker ambiguous supervision is cheaper to obtain than unambiguous supervision, it is reasonable to expect availability of higher amounts of ambiguous training data than unambiguous training data in practice.

Although artificially-constructed ambiguous training data was used in our experiments, the results indicate a promising potential for using KRISPER to learn language semantics from real-world ambiguous training data.

Conclusions

We have presented a method for learning a semantic parser when the available training data is ambiguous. Ambiguous supervision of this form is more representative of a "natural" training environment for a language-learning system. Our method learns from ambiguous data by iteratively retraining an existing learning system that utilizes unambiguous training data. Experimental results on two corpora in which ambiguities were artificially introduced demonstrate that our method is able to learn accurate semantic parsers even when supervision is ambiguous. In the future, we plan to acquire real-world corpora with natural ambiguities and test our method on them.

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

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

Cormen, T. H.; Leiserson, C. E.; and Rivest, R. L. 1990. *Introduction to Algorithms*. Cambridge, MA: MIT Press.

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

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

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

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

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

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

Munkres, J. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics* 5(1):32–38.

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

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

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