Unpublished Technical Note, January 1997

Semantic Lexicon Acquisition for Learning Parsers

Cynthia A. Thompson and Raymond J. Mooney
Department of Computer Sciences
University of Texas
Austin, TX 78712
cthomp@cs.utexas.edu, mooney@cs.utexas.edu

Abstract

This paper describes a system, WOLFIE (WOrd Learning From Interpreted Examples), that learns a semantic lexicon from a corpus of sentences paired with representations of their meaning. The lexicon learned consists of words paired with representations of their meaning, and allows for both synonymy and polysemy. WOLFIE is part of an integrated system that learns to parse novel sentences into their meaning representations. Experimental results are presented that demonstrate WOLFIE's ability to learn useful lexicons for a realistic domain. The lexicons learned by WOLFIE are also compared to those learned by another lexical acquisition system, that of [Siskind, 1996].

Introduction

There is increasing interest in automating the process of building natural language processing (NLP) systems using training corpora. The semantic lexicon, or the mapping from words to meanings, is one component that is typically difficult to construct and update, and changes from one domain to the next. Constructing a lexicon by hand is difficult and time consuming, as noted by [Copestake, 1995] and [Walker and Amsler, 1986]. Also, [Johnston *et al.*, 1995] discuss the need for systems that can learn the meanings of novel words. Therefore, automating the acquisition of the semantic lexicon is an important task in automating the development of NLP systems. This paper describes a system, WOLFIE (WOrd Learning From Interpreted Examples), that learns a semantic lexicon from input consisting of sentences paired with representations of their meanings.

Although a few others [Siskind, 1996; Hastings and Lytinen, 1994; Brent, 1991] have presented systems for lexical acquisition, this work is unique in combining several features. First, arbitrary amounts of both polysemy and synonymy can be handled. Second, interaction with a system, CHILL [Zelle, 1995], that learns to parse database queries directly into logical form is demonstrated. Third, it uses a fairly simple batch, greedy algorithm that is quite fast and accurate.

The system makes only a few fairly straightforward assumptions about the problem, some of which will be removed in future work. First is *compositionality*, i.e., the meaning of a sentence is composed from possible meanings of words and phrases in that sentence. Second, the sentence representations contain no noise. Third, the meaning for each occurrence of a word in a sentence appears only once in the sentence's representation. The second and third of these assumptions are preliminary, and methods for removing them will be investigated in future work. In the following, we will use *phrase* to refer to phrases of one or more words.

WOLFIE has been tested on acquiring a semantic lexicon for the task of answering geographical database queries using a corpus of queries collected from human subjects and annotated with their executable logical form. In this process, it has been integrated with CHILL which learns parsers but requires a semantic lexicon (previously built manually). Results demonstrate that the final application system performs only slightly worse at accurately answering questions when using a learned lexicon compared to a correct hand-built one. The system is also compared to an alternative lexical acquisition system developed by [Siskind, 1994; Siskind, 1996], demonstrating superior performance on this task.

CHILL and the Geoquery Domain

The output produced by WOLFIE can be used to assist a larger language acquisition system; in particular, it is currently used as part of the input to CHILL, a parser acquisition system. CHILL learns parsers from a corpus of sentences paired with their semantic meanings, the same type of corpus required by WOLFIE. Currently, CHILL requires a lexicon as background knowledge in order to learn to parse into deep semantic representations. By using WOLFIE, one of the inputs to CHILL is automatically provided, thus easing the task of parser acquisition.

In this paper, we will limit our discussion of Chill to its ability to learn parsers that can map natural language database queries about geography directly into an executable Prolog query that answers the question [Zelle and Mooney, 1996]. Following are two examples of sentences for this domain, the *geoquery* domain, paired with their corresponding Prolog query:

What is the capital of the state with the biggest population?
```
answer(C, (capital(S,C), largest(P,
   (state(S), population(S,P))))).
```
What state is texarkana located in?
```
answer(S, (state(S),
   eq(C,cityid(texarkana,_)), loc(C,S))).
```

Given a corpus of sentence-representation pairs in this format, Chill is able to learn a parser that can parse novel sentences into the database query format. Wolfie assists Chill by learning mappings between words and the predicates and terms in these queries.

The Lexical Learning Problem

We now define the Lexical Learning Problem solved by Wolfie, after introducing some definitions and terminology. Let S be $\{s_1, s_2, \ldots, s_n\}$, a set of sentences each containing an ordered list of words. We will denote the list for the ith sentence, containing m words, as $(w_{i_1}, w_{i_2}, \ldots, w_{i_m})$. Let R be $\{r_1, r_2, \ldots, r_n\}$, a set of meaning representations for the corresponding sentences in S; and I be $\{(s_1, r_1), (s_2, r_2), \ldots, (s_n, r_n)\}$, a set of *(sentence, representation)* pairs drawn from S and R. An element of R can be `fracture`d into all of its subcomponents, denoted p_j, and the method for doing so depends upon R. [Siskind, 1992] was the first to utilize this notion of fracturing within a lexical learning procedure. For each valid set of these subcomponents, we can build them back into a valid sentence meaning using a relation we will call `compose`.

The goal of Lexical Learning is to find a semantic lexicon that will simplify both parsing and the acquisition of parsers. The learner is given I as input, and either an implementation of `fracture` is given or it is implicit in the learning algorithm. The goal is to find a lexicon, M, of *(phrase, meaning)* pairs, where the phrases and their meanings are extracted from the sentences and their representations, respectively, and for each pair $(s_i, r_i) \in I$, where $s_i = (w_{i_1}, w_{i_2}, \ldots, w_{i_m})$, it must be the case that we can choose a set of j pairs from M, $1 \leq j \leq m$, where each pair is of the form $([w_{i_k}, w_{i_{k+1}}, \ldots, w_{i_{k+q}}], p_j)$, $1 \leq k \leq k+q \leq m$. The phrases in one pair of the set do not intersect the phrases in any other, and `compose`$(\{p_1, p_2, \ldots, p_j\}, r_i)$ must be valid. In other words, each sentence's representation can be `compose`d from the possible meanings of the (unique) phrases in the sentence. Ideally, we would like to minimize the size and ambiguity of the learned lexicon, since we hypothesize that this will ease the parser acquisition task for Chill. Therefore, our learner will also try to minimize these.

We make no assumption that each phrase has a single meaning (i.e., homonymy is allowed), or that each meaning is associated with one phrase only (i.e., synonymy is allowed). Also, some phrases in S may have a null meaning associated with them.

Such learning is possible under our compositionality assumption. This allows the initially hypothesized meanings of a phrase to consist of pieces of the representations of sentences in which the phrase appears. To look for the best meaning for a phrase, one could collect the fractured representations of all sentences in which the phrase appears. Then, the one with the best coverage for that phrase could be chosen as the meaning for the phrase.

This method would be computationally expensive and would not take advantage of the constraints between phrase meanings. Such constraints exist because of our assumption that each portion of the representation is due to only one phrase in the sentence. Therefore, once part of a sentence's representation is covered by the meaning of one of the phrases in the sentence, we know that no other phrases in the sentence can have that meaning, for that sentence. This will be illustrated more clearly in the next section.

The Wolfie Algorithm and an Example

In order to limit the search for phrase meanings, a greedy algorithm is used. At each step, the best phrase-meaning pair is chosen, according to a heuristic described below, and added to the lexicon. The list of potential meanings for a phrase is formed by fracturing the representation of the sentences in which the phrase appears and computing the common substructure between sampled pairs of the resulting components.

One of the key ideas of the algorithm is that each choice of a lexical item may constrain the possible meanings of phrases not yet learned. This can best be illustrated by an example. Let us assume we have the sentence-representation pairs given in the previous column, plus the additional pair:

What is the highest point of the state with the biggest area?
```
answer(P, (high-point(S,P), largest(A,
   (state(S), area(S,A))))).
```

Let us assume for the purpose of this example that

```
Derive possible phrase-meaning pairs by sampling
   the input sentence-representation pairs that
   have phrases in common, and deriving the common
   substructure in their representations.
Until all input representations can be composed
   from their phrase meanings do:
      Add the best phrase-meaning pair to the lexicon.
      Constrain the remaining possible phrase-meaning
         pairs to reflect the pair just learned.
Return the lexicon of learned phrase-meaning pairs.
```

Figure 1: Wolfie Algorithm Overview

we strip each sentence of phrases that we know, *a priori*, will have a null meaning representation in all sentences. In the example sentences, those phrases would be [**what**], [**is**], and [**the**].

From these three sentences, then, the meaning of [**state**], the only phrase common to all sentences, is uniquely determined as `state(_)`, which is the only predicate the three representations have in common[1]. Before determining this, the list of potential meanings for [**biggest**] is [`largest(_, state(_))`]. However, since `state(_)` is now covered by [**state**], it can be eliminated from consideration as part of the meaning of [**biggest**], and the list of potential meanings for [**biggest**] becomes [`largest(_,_)`].

The Wolfie algorithm, outlined in Figure 1, has been implemented to handle sentences paired with two kinds of representations. First, it can handle sentences with a case-role semantic representation, such as *conceptual dependency* [Schank, 1975]. For example, the sentence "The man ate the cheese" is represented by *[ingest, agent: [person, sex:male, age:adult], patient: [food, type:cheese]]*.

The second representation handled is the logical query representation used in the geoquery domain, the one focussed on in the remainder of this paper. To find the common substructure between pairs of query representations, we use a method which is similar to that of finding Least General Generalizations (LGGs) of clauses [Plotkin, 1970]. To summarize, the LGG of two clauses is the least general clause that subsumes both clauses. For example, given the queries from the previous page, the common substructure is `state(_)`.

We now describe the algorithm in more detail. The first step of the algorithm derives common substructure for a random sample of one and two-word phrases in the corpus. Future work includes the ability to handle longer phrases. For example, let us suppose the following pairs as input:

What is the capital of the state with the biggest population?
```
answer(C, (capital(S,C), largest(P,
   (state(S), population(S,P))))).
```
What is the highest point of the state with the biggest area?
```
answer(P, (high-point(S,P), largest(A,
   (state(S), area(S,A))))).
```
What state is texarkana located in?
```
answer(S, (state(S),
   eq(C,cityid(texarkana,_)), loc(C,S))).
```
What is the area of the united states?
```
answer(A, (area(C,A),
   eq(C,countryid(usa)))).
```
What is the population of the states bordering minnesota?
```
answer(P, (population(S, P), state(S),
   next_to(S,M),
   eq(M,stateid(minnesota)))).
```

The sets of initial potential meanings for some of the words in this corpus are:
[**biggest**]: [`largest(_,state(_))`],
[**state**]: [`state(_)`,`largest(_,state(_))`],
[**area**]: [`area(_)`],
[**population**]: [`(population(_,_), state(_))`],
[**capital**]: [`(capital(S,_), largest(P, (state(S), population(S,P))))`].

After deriving these initial meanings, the greedy search begins. At each step of the search, the best phrase-meaning pair is added to the lexicon. We use a heuristic to estimate the value of each phrase-meaning pair. The heuristic has five weighted components as follows:

1. Ratio of the number of times the phrase appears with the meaning to the number of times the phrase appears, or *coverage*.

2. Ratio of the number of times the phrase appears with the meaning to the number of times the meaning appears.

3. Percent of orthographic overlap between the phrase and its meaning.

4. Percent rank of the phrase's frequency out of all phrases in the corpus.

5. The generality of the meaning.

The intuition behind the first measure is that high coverage should help lead to a small lexicon. The second measure helps reduce the ambiguity of the learned lexicon. The third measure is used for this corpus since often phrases have many characters in common with

[1] Since Chill initializes every parse with the `answer/2` predicate, it is first stripped from the input given to Wolfie.

their meanings, as in **area** and `area(_)`. It measures the maximum number of consecutive characters in common between the phrase and the terms and predicates in the meanings, as an average of the percent of both the number of characters in the phrase and in the term and predicate names. Fourth, common words are more likely to be paired with a correct meaning, so preferring them should lead to a better learned lexicon. The final measure, generality, measures the number of terms and predicates in the meaning. Learning a phrase meaning with fewer terms should lead to a lexicon with less ambiguity.

For purposes of this example, we will use a weight of 25 for each of the first four parameters, and a weight of one for the last. Also, we will assume that closed class words, such as determiners, and state and city names, are known in advance and thus removed from the input pairs. Tie breakers between two phrases are learning less "ambiguous" phrases first and learning phrases with fewer words before phrases with more words. We count a phrase as more ambiguous than another if there are more meanings in the learned lexicon for it than the other. The assumption here is that a less ambiguous lexicon will make for a better lexicon for parsing.

The heuristic measure for the above five pairs is:
[[**biggest**], `largest(_,state(_))`]:
$25(2/2)+25(2/2)+25((4/12+4/7)/2)+25(2/20)-1*2 = 61.8$,
[[**area**], `area(_)`]:
$25(2/2)+25(2/2)+25((4/4+4/4)/2)+25(2/20)-1*1 = 76.5$,
[[**state**], `state(_)`]:
$25(3/3)+25(3/4)+25((5/5+5/5)/2)+25(3/20)-1*1 = 71.5$,
[[**state**], `largest(_,state(_))`]: 61.2,
[[**population**], `(population(_,_), state(_))`]: 71.3,
[[**capital**], `(capital(S,_), largest(P, (state(S), population(S,P))))`]: 62.8.
The best pair by our measure is [[**area**], `area(_)`].

The next step in the algorithm is to constrain the remaining hypothesized meanings, if any, for the learned phrase, so as to only consider sentences for which no meaning has yet been learned for the phrase. In our example, the learned pair covers all occurrences of [**area**]. Next, for the remaining unlearned phrases, their hypothesized meanings are constrained to take into account the meaning just learned. In our example, learning [**area**] would not effect any of the listed meanings, but the next best pair, [[**state**], `state(_)`], would constrain the meaning for [**population**] to become `population(_,_)`. The greedy search continues until all representations in the input can be assembled from learned phrase meanings.

Experimental Results

Experiments were performed measuring the usefulness of the semantic lexicons learned by Wolfie as background knowledge for Chill. Wolfie was trained on subsets of the geoquery corpus, the learned lexicons were used as background knowledge when training Chill on the same subsets, and the resulting parser was evaluated on the unseen examples. In addition, the ability of the learned lexicons to cover the testing corpus was measured as described below. The corpus consists of 250 sentences.

We compared our system to that of [Siskind, 1996]. To use his system, the input had to be slightly modified. The representation
`largest(P,(capital(C),population(C,P)))` was changed to
`((largest2 population (and (capital1 capital) (population2 capital population))))`. The numbers on predicate names are used to distinguish them from the variables. Tokens were used in lieu of variables, since his system does not accept representations with variables in predicate arguments. Closed class words and state and city names were also removed from the training corpora given to Siskind's system.

A random set of training examples was chosen, starting with 25 examples, and incrementing by 50, for each of 5 trials. We used a weight of one for the generality measure of the heuristic function. To intelligently choose the weights of the first four parameters for the heuristic, we used 10-fold cross-validation [Kohavi and John, 1995]. 15 different predefined weight sets were evaluated, in addition to four random weight sets[2]. To choose the best weight set, we measured the coverage of the learned lexicon for the held out training sentences, and the ambiguity, A, of the lexicon. Coverage of a sentence-representation pair was measured by the percent of the terms and predicates in its representation that could be covered by learned meanings of the words in its sentence. Ambiguity was simply the total number of phrases in the lexicon divided by the number of unique phrases in the lexicon. The weight set that resulted in the highest value when averaging coverage and $1/A$ was chosen to learn a lexicon from the full set of training examples.

Since Siskind has no measure of orthographic overlap, and it could arguably give our system an unfair advantage in this domain, we ran a second set of tests without using this part of the heuristic. The weights of the other heuristics were again chosen by cross-validation.

Figure 2 shows the results of using the lexicons

[2] We are currently working on the implementation of a version which uses a best first search for the weight sets, as in [Kohavi and John, 1995]

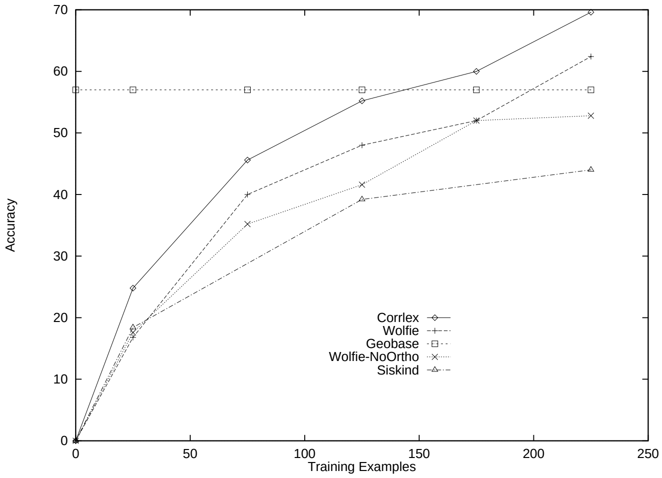

Figure 2: Accuracy of CHILL with Learned vs. Correct Lexicons

System	125	225
Wolfie	91.7%	97.3%
Wolfie-NoOrtho	83.2%	91.2%
Siskind	82.6%	89.5%

Table 1: Coverage Results

learned by the two systems as background knowledge for CHILL. The figure shows accuracy of the parsers learned by CHILL with the learned and a hand-generated lexicon as background knowledge. The accuracy is the percentage of test sentences for which the correct answer to the query was produced. The horizontal line is the accuracy for a hand-built application, *Geobase*, supplied with a commercial Prolog system, Turbo Prolog 2.0 [Borland International, 1988].

The results show that a lexicon learned by WOLFIE with cross-validation considering all heuristic measures led to learned parsers that were slightly worse than parsers learned from the hand-built lexicon. The results at 25 examples were mixed, but for higher numbers of training examples, the best accuracy is the hand-built lexicon, followed by WOLFIE with all heuristics considered, followed by WOLFIE without orthographic overlap, followed by Siskind's system. All the systems except Siskind's and WOLFIE without orthographic overlap do better than *Geobase* after 225 training examples.

These results show that WOLFIE can learn lexicons that lead to successful learning of parsers by CHILL. Though not as valid a metric as the accuracy of learned parsers, the ability of the learned lexicons to cover the testing sentences is also useful to examine. This is the same notion of coverage as was used in the cross-validation experiment described previously. Table 1 shows these percentages. *Wolfie* is WOLFIE using all heuristics in the cross-validation, *Wolfie-NoOrtho* is WOLFIE without orthographic overlap, and the third line is Siskind's system. Both versions of WOLFIE have better coverage than Siskind's system.

The weight sets chosen by the cross-validation method were not consistent across the five runs. However, the first measure, coverage, was given at least a weight of 25 on all of the runs with 175 examples or more.

Future Work

The results to date are promising, but we intend to continue to improve the system and perform additional experiments. Enhancing the search heuristic and expanding the search may improve the learned lexicons. Active learning, where the system chooses which examples can be most usefully annotated, will be examined. We also plan to investigate the use of background knowledge, such as WordNet [Beckwith *et al.*, 1991], by adding to the heuristic a preference for matching a word to terms in the representation that are semantically related. Methods for handling noisy data are also needed. Additional comparisons using the current corpus are needed to establish statistical significance. Also, we are currently in the process of running experiments on a version of the geoquery corpus relabelled in Spanish, and constructing corpora of natural language queries about jobs from information extracted from the newsgroup `misc.jobs.offerred`.

Related Work

A method for acquiring syntactic and semantic features of an unknown word is described by [Pedersen and Chen, 1995]. They assume access to an initial concept hierarchy, and show no experimental results. Many systems [Fukumoto and Tsujii, 1995; Haruno, 1995; Johnston *et al.*, 1995] focus only on the acquisition of verbs or nouns, rather than both. [Manning, 1993; Brent, 1991] acquire subcategorization information for verbs.

The most closely related work is that of Siskind. His system handles some situations that ours cannot. For example, it handles noise and referential uncertainty (multiple possible meanings for a sentence). While his system is more general in this sense, our system is specialized for applications where a single correct meaning for each sentence can be given. His system does not currently handle multiple-word phrases. Also, his system operates in an incremental or on-line fashion, discarding each sentence as it processes it, while ours is batch. While he argues for psychological plausibility, we do not.

Conclusion

Acquiring a semantic lexicon from a corpus of sentences labelled with representations of their meaning is an important problem that has not been widely studied. Wolfie demonstrates that a fairly simple greedy symbolic learning algorithm performs fairly well on this task and superior to a previous lexicon acquisition system on a corpus of geography queries.

Most experiments in corpus-based natural language have presented results on some subtask of natural language, and there are little if any results on whether the learned subsystems can be successfully integrated to build a complete system. The experiments presented in this paper demonstrate how two learning systems, Wolfie and Chill can be successfully integrated to learn a complete NLP system for parsing database queries into executable logical form given only a corpus of annotated queries.

Acknowledgements

We would like to thank J. Siskind for providing us with his software, and for all his help in adapting it for use with our corpus. This research was supported by the National Science Foundation under grant IRI-9102926.

References

[Beckwith *et al.*, 1991] Beckwith, R.; Fellbaum, C.; Gross, D.; and Miller, G. 1991. Wordnet: A lexical database organized on psycholinguistic principles. In Zernik, U., editor 1991, *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon.* Lawrence Erlbaum, Hillsdale, NJ. 211–232.

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

[Brent, 1991] Brent, M. 1991. Automatic acquisition of subcategorization frames from untagged text. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics.* 209–214.

[Copestake, 1995] Copestake, et. al. A. 1995. Acquisition of lexical translation relations from MRDS. *Machine Translation* 9.

[Fukumoto and Tsujii, 1995] Fukumoto, Fumiyo and Tsujii, Jun'ichi 1995. Representation and acquisition of verbal polysemy. In *Papers from the 1995 AAAI Symposium on the Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, Stanford, CA. 39–44.

[Haruno, 1995] Haruno, Masahiko 1995. A case frame learning method for Japanese polysemous verbs. In *Papers from the 1995 AAAI Symposium on the Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, Stanford, CA. 45–50.

[Hastings and Lytinen, 1994] Hastings, P. and Lytinen, Steven 1994. The ups and downs of lexical acquisition. In *Proceedings of the Twelfth National Conference on Artificial Intelligence.* 754–759.

[Johnston *et al.*, 1995] Johnston, M.; Boguraev, B.; and Pustejovsky, J. 1995. The acquisition and interpretation of complex nominals. In *Papers from the 1995 AAAI Symposium on the Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, Stanford, CA. 69–74.

[Kohavi and John, 1995] Kohavi, R. and John, G. 1995. Automatic parameter selection by minimizing estimated error. In *Proceedings of the Twelfth International Conference on Machine Lea rning*, Tahoe City, CA. 304–312.

[Manning, 1993] Manning, Christopher D. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio. 235–242.

[Pedersen and Chen, 1995] Pedersen, Ted and Chen, Weidong 1995. Lexical acquisition via constraint solving. In *Papers from the 1995 AAAI Symposium on the Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, Stanford, CA. 118–122.

[Plotkin, 1970] Plotkin, G. D. 1970. A note on inductive generalization. In Meltzer, B. and Michie, D., editors 1970, *Machine Intelligence (Vol. 5).* Elsevier North-Holland, New York.

[Schank, 1975] Schank, R. C. 1975. *Conceptual Information Processing.* North-Holland, Oxford.

[Siskind, 1992] Siskind, Jeffrey M. 1992. *Naive Physics, Event Perception, Lexical Semantics and Language Acquisition.* Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

[Siskind, 1994] Siskind, Jeffrey M. 1994. Lexical acquisition in the presence of noise and homonymy. In *Proceedings of the Twelfth National Conference on Artificial Intelligence.* 760–766.

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

[Walker and Amsler, 1986] Walker, D. and Amsler, R. 1986. The use of machine-readable dictionaries in sublanguage analysis. In Grishman, R. and Kittredge, R., editors 1986, *Analyzing Language in Restricted Domains.* Lawrence Erlbaum Associates, Hillsdale, NJ. 69–83.

[Zelle and Mooney, 1996] Zelle, J. M. and Mooney, R. J. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, OR.

[Zelle, 1995] Zelle, J. M. 1995. *Using Inductive Logic Programming to Automate the Construction of Natural Language Parsers.* Ph.D. Dissertation, University of Texas, Austin, TX. Also appears as Artificial Intelligence Laboratory Technical Report AI 96-249.