Incorporating Knowledge in Natural Language Learning: A Case Study

Yuval Krymolowski
Dept. of Math and Computer Science
Bar-Ilan University
52900 Ramat Gan, Israel
yuvalk@cs.biu.ac.il

Dan Roth
Dept. of Computer Science
University of Illinois
Urbana, IL 61801
danr@cs.uiuc.edu

Abstract

Incorporating external information during a learning process is expected to improve its efficiency. We study a method for incorporating noun-class information, in the context of learning to resolve Prepositional Phrase Attachment (PPA) disambiguation. This is done within a recently introduced architecture, *SNOW*, a sparse network of threshold gates utilizing the Winnow learning algorithm. That architecture has already been demonstrated to perform remarkably well on a number of natural language learning tasks.

The knowledge sources used were compiled from the WordNet database for general linguistic purposes, irrespective of the PPA problem, and are being incorporated into the learning algorithm by enriching its feature space. We study two strategies of using enriched features and the effects of using class information at different granularities, as well as randomly-generated knowledge which serves as a control set.

Incorporating external knowledge sources within *SNOW* yields a statistically significant performance improvement. In addition, we find an interesting relation between the granularity of the knowledge sources used and the magnitude of the improvement. The encouraging results with noun-class data provide a motivation for carrying out more work on generating better linguistic knowledge sources.

1 Introduction

A variety of inductive learning techniques have been used in recent years in natural language processing. Given a large training corpus as input and relying on statistical properties of language usage, statistics-based and machine learning algorithms are used to induce a classifier which can be used to resolve a disambiguation task. Applications of this line of research include ambiguity resolution at different levels of sentence analysis: part-of speech tagging, word-sense disambiguation, word selection in machine translation, context-sensitive spelling correction, word selection in speech recognition, and identification of discourse markers.

Many natural language inferences, however, seem to rely heavily on semantic and pragmatic knowledge about the world and the language, that is not explicit in the training data. The ability to incorporate knowledge from other sources of information, be it knowledge that is acquired across modalities, prepared by a teacher or by an expert, is crucial for going beyond low level natural language inferences.

Within Machine Learning, the use of knowledge is often limited to that of constraining the hypothesis space (either before learning or by probabilistically biasing the search for the hypothesis) or to techniques such as EBL (DeJong, 1981; Mitchell et al., 1986; DeJong and Mooney, 1986) which rely on explicit domain knowledge that can be used to *explain* (usually, prove deductively) the observed examples.

The knowledge needed to perform language-understanding related tasks, however, does not exist in any explicit form that is amenable to techniques of this sort, and many believe that it will never be available in such explicit forms. An enormous amount of useful "knowledge" may be available, though. Pieces of information that may be found valuable in language-understanding related tasks may include: the root form of a verb; a list of nouns that are in some relation (e.g., are all countries) and can thus appear in similar contexts; a list of verbs that can be followed by a food item; a list of items you can see through, things that are furniture, a list of dangerous things, etc.

This rich collection of information pieces does not form any domain theory to speak of and cannot be acquired from a single source of information. This knowledge is noisy, incomplete and ambiguous. While some of it may be acquired from text, a lot if it may only be acquired from other modalities, as those used by humans. We believe that integration of such knowledge is essential for NLP to attain high-level natural-language inference.

Contrary to this intuition, experiments in text retrieval and natural language have not shown much improvement when incorporating information of the kind humans seem to use (Krovetz and Croft, 1992; Kosmynin and Davidson, 1996; Karov and Edelman, 1996; Junker, 1997). The lack of significant improvement in the presence of more "knowledge" may be explained by the type of knowledge used, the way it is incorporated, and the learning algorithms employed.

In the present paper we study an effective way of incorporating incomplete and ambiguous information sources of the abovementioned type within a specific learning approach, and focus on the knowledge sources that can be effective in doing so. The long-term goal of our work is understanding (1) what types of knowledge sources can be used for performance improvement, and at what granularity level and (2) which computational mechanisms can make the best use of these sources.

In particular, the effect of noun-class information on learning Prepositional Phrase Attachment (PPA, cf. Sec. 2) is studied. This problem is studied within *SNOW*, a sparse architecture utilizing an on-line learning algorithm based on Winnow (Littlestone, 1988). That algorithm has been applied for natural language disambiguation tasks and related problems and perform remarkably well (Golding and Roth, 1996; Dagan et al., 1997; Roth and Zelenko, 1998).

The noun-class data was derived from the WordNet database (Miller, 1990) which was compiled for general linguistic purposes, irrespective of the PPA problem. We derived the classes at different granularities. At the highest level, nouns are classified according to their synsets. The lower levels are obtained by successively using the hypernym relation defined in WordNet. In addition, we use the *Corelex* database (Buitelaar, 1998). Consisting of 126 coarse-grained semantic types covering around 40,000 nouns, *Corelex* defines a large number of systematic polysemous classes that are derived from an analysis of sense distributions in WordNet.

The results indicate that a statistically significant improvement in performance is achieved when the noun-class information is incorporated into the data. The absolute performance achieved on the task is slightly better than other systems, although it is still significantly worse than the performance of a human subject tested on this task. The granularity of the class information appears to be crucial for improving performance. The addition of too many overlapping classes does not help performance, but with fewer classes - the improvement is significant.

In addition to semantic information, using classes carries with it some structural information. A class feature may be viewed as a disjunction of other features, thereby increasing the expressivity of the hypothesis used for prediction. In order to control for the possibility that the performance improvements seen are due mainly to the structural information, we generated random classes. Some of these had exactly the same distribution over the original features as do the semantic classes. Surprisingly, we find that a non-negligible part of the improvement is due merely to the structural information, although most of it can be attributed to the semantic content of the classes.

Along with promoting work on the incorporation of problem-independent incomplete knowledge into the learning process, the encouraging results with incorporating noun-class data provide a motivation for carrying out more work on generating better linguistic knowledge sources.

The paper is organized as follows: we start by presenting the task, PPA and the *SNOW* architecture and algorithm. In section 4 we describe the classes and present the main experiments with the semantic and random classes. Section 5 concludes.

2 Prepositional phrase attachment

The PPA problem is to decide whether the prepositional phrase (PP) attaches to the direct object NP as in `Buy the car with the steering wheel` (n-attachment) or to the verb phrase `buy`, as in `Buy the car with his money` (v-attachment). PPA is a common cause of structural ambiguity in natural language.

Earlier works on this problem (Ratnaparkhi et al., 1994; Brill and Resnik, 1994; Collins and Brooks, 1995; Zavrel et al., 1997) represented an example by the 4-tuple <v, n1, p, n2> containing the VP head, the direct object NP head, the preposition, and the indirect object NP head respectively. The first example in the previous paragraph is thus represented by <`buy, car, with, wheel`>.

The experiments reported here were done using data extracted by Ratnaparkhi et al. (1994) from the Penn Treebank (Marcus et al., 1993) WSJ corpus. It consists of 20801 training examples and 3097 separate test examples.

The preposition `of` turns out to be a very strong indicator for noun attachment. Among the 3097 test examples, 925 contain the preposition `of`; in all but 9 of these examples, `of` has an `n` attachment.

Since almost all (99.1%) of these test cases are classified correctly regardless of the *SNOW* architecture or parameter choice, we omit the examples which include `of` from the test set, as they obscure the real performance. Only the last table will include those examples, so results may be compared with other systems evaluated on this data set.

In summary, our data set consists of 15224 training examples, (5338 tagged `n`, 9886 tagged `v`) and 2172 test examples (910 and 1262, resp.). This leads to a baseline performance of 58.1% if we simply predict according to the most common attachment in the training corpus: `v`. (Simply breaking this down to different prepositions does not yield better results.) For reference, assuming a binomial distribution, the standard deviation on the test set is 0.85%. That figure is a crude estimator of the standard deviation of the results.

A study of the possible features which may be extracted from the data, shows that the best feature set is that composed of all the possible conjunctions of words in the input 4-tuple. In addition, lemmatizing all the nouns and verbs yielded a further performance improvement. In the following section we will use the lemmatized data "`lemma`" as a basic set.

3 The *SNOW* Approach

The *SNOW* architecture is a network of threshold gates. Nodes in the first layer of the network represent the input features; target nodes are represented by nodes in the second layer. Links from the first to the second layer have weights; each target node is thus defined as a (linear) function of the lower level nodes.

For example, in PPA, the two target nodes represent `n` and `v` attachments. Each target node can be thought of as an autonomous subnetwork, although they all feed from the same input. The subnetworks are *sparse* in that a target node needs not be connected to all nodes in the input layer. For example, it is not connected to input nodes (features) that were never active with it in the same example, or it may disconnect itself from some of the irrelevant inputs while training.

Learning in *SNOW* proceeds in an on-line fashion[1]. Every example is treated autonomously by each target subnetwork, viewed as a positive example of a few subnetworks and a negative example for the others. In PPA, examples labeled `n` (`v`, resp.) are treated as positive for the `n` (`v`) target node and as negative for the `v` (`n`) target node. Thus, every example is used once by all the nodes to refine their definition, and then discarded. At prediction time, given an input which activates a subset of the input nodes, each subnetwork evaluates the total activity it receives. Subnetworks compete on determining the final prediction; the one which produces the highest activity gets to determine the prediction.

In general, a target node in the *SNOW* architecture is represented by a collection of subnetworks, which we call a *cloud*, but in the application described here we have used cloud size of 1 so this will not be discussed here.

The Winnow local mistake-driven learning algorithm (Littlestone, 1988) is used at each target node to learn its dependence on the input nodes. Winnow updates the weight on the links in a multiplicative fashion. We do not supply the details of the algorithm and just note that it can be implemented in such a way that the update time of the algorithm depends on the number of *active* features in the example rather than the *total* number of features in the domain. The sparse architecture along with the representation of each example as a list of active features is reminiscent of infinite attribute models of Winnow (Blum, 1992).

Theoretical analysis has shown that multiplicative update algorithms, like Winnow, have exceptionally good behavior in the presence of irrelevant attributes, noise, and even a target function changing in time (Littlestone, 1988; Littlestone and Warmuth, 1994; Herbster and Warmuth, 1995). In particular, Winnow was shown to learn efficiently any linear threshold function (Littlestone, 1988), with a mistake bound that depends on the margin between positive and negative examples. The key feature of Winnow is that its mistake bound grows linearly with the number of *relevant* attributes and only logarithmically with the total number of attributes n. In particular, Winnow still maintains its abovementioned dependence on the number of total and relevant attributes even when no linear-threshold function can make a perfect classification (Littlestone, 1991; Kivinen and Warmuth, 1995).

Even when there are only two target nodes and the cloud size is 1, the behavior of *SNOW* is different from that of pure Winnow. While each of the target nodes is learned using a positive Winnow algorithm, a winner-take-all policy is used to determine the prediction. Thus, we use the learning algorithm here in a more complex way than just as a discriminator. One reason is that the *SNOW* architecture, influenced by the Neuroidal system (Valiant, 1994), is being used in a system developed for the purpose of learning knowledge representations for natural language understanding tasks, and is being evaluated on a variety of tasks for which the node allocation process is of importance.

We have experimented extensively with various architectures of *SNOW* on the PPA problem but can present in this paper only a small part of these experiments. The best performance, across a few parameter sets and data, is achieved with a *full* architecture. In this case we initially link a target node to *all* features which occur in the training (with a constant initial value), and only then start training. Since training in *SNOW* is always done in an on-line fashion - each example is used only once for updating the weights, and only if a mistake on it was made.

4 Incorporating Semantic Knowledge

In this section we describe the effect of incorporating semantic knowledge on learning PPA with *SNOW*. The information sources are briefly described in Sec. 4.1, the experimental results are reported in

[1] In the experimental study we do not update the network while testing.

Sec. 4.2, and results with random classes, used as a control set, are presented in Sec. 4.3.

Winnow has three parameters: a threshold θ and two update parameters, a *promotion* parameter $\alpha > 1$ and a *demotion* parameter $0 < \beta < 1$. The experiments reported here were made using the full *SNOW* architecture, with $\beta = 0.85$, $\alpha = \frac{1}{\beta}$, $\theta = 1$, and all the weights initialized to 0.1.

4.1 Semantic Data Sources

The semantic data sources specify for each noun a set of semantic classes. These classes result from a general linguistic study, hence not biased so as to present data in the context of PPA. In addition, the vocabularies which the semantic data cover overlaps our train and test data vocabulary only partially. Table 1 shows a summary of the class data. The knowledge sources which were incorporated are:

WordNet(WN): WordNet–1.6 noun class information was used at various granularity levels. In the highest level, denoted by WN1, nouns are classified according to their synsets. The lower levels are obtained by successively using the hypernym relation defined in WordNet. Thus, WN2 is obtained by replacing each WN1 synset with the set of hypernyms to which it points, WN3 – by performing a similar process on the WN2 hypernyms, etc. We have used WN1, WN5, WN10, and WN15, Table 1 lists properties of these datasets.

CoreLex(CL): The *Corelex* database (Buitelaar, 1998) was derived from WordNet as part of a linguistic research attempting to provide a unified approach to the systematic polysemy and underspecification of nouns. Systematic polysemy is the phenomena of word senses that are systematically related and therefore predictable over classes of lexical items. The thesis behind this data base is that acknowledging the systematic nature of polysemy allows one to structure ontologies for lexical semantic processing that may help in generating more appropriate interpretations within context. The data base establishes an ontology and semantic database of 126 semantic types, covering around 40,000 nouns that were derived by an analysis of sense distributions in WordNet.

It is clear that with such a coarse-grained ontology, a lot of information is being lost. This is a many-to-one mapping in which many words fall into a class due only to one of their senses, and there are cases of incomplete and inaccurate information. For example, `observatory` falls into the class of `artifact state`; words like `dog, lion, table` are missing from the vocabulary.

Format Features (FF): These are two classes into which one can classify nouns using simple heuristics. The first consists of numbers (e.g., $1, 2, 100$, `three`, `million`), and the second contains proper nouns. Each noun beginning with a capital letter was classified as a proper noun, which clearly gives a very crude approximation.

4.2 Experimental Results

In this section we present results of incorporating various semantic data and their combinations. Since the classes were not compiled specifically for the PPA problem, some of the class information may be irrelevant or even slightly misleading. The results provide an assesment of the relative relevance of each knowledge source.

When a noun belongs to a class, one may *replace* the explicit noun feature by its classes. Using the classes *in addition* to the original noun (Brill and Resnik, 1994; Resnik, 1992; Resnik, 1995) seems, however, a better strategy. Consider, for example, the feature <`prep`,indirect-object=`n2`>. Suppose the noun `n2` belongs to two classes `c1` and `c2`. The class information will be incorporated by creating two *additional* features: <`prep`,indirect-object=`c1`> and <`prep`,indirect-object=`c2`>, thereby enhancing the feature set without losing the original information. As mentioned above, giving up the original feature yielded degraded results.

The results of adding features from a single knowledge source, presented in Table 2, show that FF have yielded small improvements over the `lemma` set, within the noise-level; the WN1 synset information caused a slight degradation, and the CL and other WN knowledge resulted in a significant improvement over the `lemma` case.

An important property of the CL class information is that each CL class defines a *distinct* set of nouns, as each noun belongs to one CL class. The synset (WN1) distribution differs greatly from that of the CL classes; each noun may belong to a few synsets – allowing more potential conflicts. That property of the synset distribution gives rise to the performance degradation.

Another important difference between CL and WN1 classes is their granularity. There are around 60000 synsets, whereas there are only 126 CL classes. The finer synset granularity means that a synset carries less information; thus, the CL classes add richer disjunctions than WN synsets do. The results of CL, WN5, WN10, and WN15 improve over the FF set, these results are within the noise level (cf. Sec. 2).

The FF set covers relatively few nouns, hence the improvement it yields is quite small. The WordNet and CL vocabularies do not include those beginning with a capital as well as numbers, therefore the WN and CL knowledge may be augmented with the FF information without loss of consistency. Nevertheless, since each number-word (e.g., "one", "two", etc.) belongs to a different synset, augmenting WN1 with a numeric class is not expected to be very effective because the words "one", "two", and "1" will all

	data	FF	CL	WN1	WN5	WN10	WN15
nouns in train	6533	2025	3083	4012	4012	4012	4012
nouns in test	1805	150	1107	1559	1559	1559	1559
nouns in both	1452	83	902	1322	1322	1322	1322
classes in train	-	2	110	10029	521	33	9
classes in test	-	2	92	5216	353	28	9
classes in both	-	2	91	4863	343	28	9

Table 1: Sizes and coverage of the noun vocabulary and classes in the various noun-class sources. The leftmost column shows the noun vocabulary size and coverage for the train and test data.

Baseline	lemma	+FF	+CL	+WN1	+WN5	+WN10	+WN15
58.1	77.4	77.8	78.6	77.2	79.1	78.5	78.6

Table 2: **Learning results for a single knowledge source:** Baseline refers to simply predicting according to the most common attachment in the training corpus, namely (`v`). `lemma` is our basic feature set, as in Sec. 2 The other columns present the prediction accuracy when adding each of our knowledge sources separately.

belong to different classes: synset(one), synset(two), and FF(is-number), respectively.

As a measure of numeric class assignment, we have examined the words: "one", "two", "three", "ten", "hundred" and "million"; only CL, WN3 and subsequent WN knowledge sources assign the same hypernym to these words, therefore we have augmented these sources.

The results are presented in Table 3, comparison with Table 2 shows that augmenting with FF knowledge yielded a slight improvement only for the CL set. There may be two explanations for that: (i) the CL classes are more appropriate for the PPA problem than the WN hypernyms, therefore the FF information fit with less conflicts. (ii) The coverage of CL nouns is about 70% that of WN for the test data (cf. Table 1), therefore there are more examples in which the CL and FF classes do not conflict. This issue requires further study.

4.3 Comparison with Random Classes

Adding semantic class information improved *SNOW* learning results. However, adding class information is equivalent to adding disjunctions of the original features and, taking aside the semantic origin of the classes, the mere introduction of disjunctions enriches the knowledge representation and may yield a performance improvement.

The motivation for using semantic classes goes, however, beyond this structural information. Nouns which haven't appeared in the training data may appear in the test data under a known class; such nouns will thus be handled based on the experience gathered for the class.

In this section we attempt to isolate the semantic content of the classes from their disjunctive meaning. Random classes, which mimic in different aspects the structure of the semantic CL classes, were constructed. Comparing the results obtained with these classes with the results using CL classes, one can see the influence of the semantic aspect of CL classes. Only some of the randomization strategies used are described here, these are:

[**CL200:**] 200 classes uniformly distributed over CL nouns.

[**CL126:**] 126 classes uniformly distributed over CL nouns. Here the number of classes in CL is maintained.

[**CL-PERM:**] A permutation of CL nouns among their classes. This random structure preserves the original class distribution of CL.

The random class results, shown in Table 4, indicate that indeed some of the gain in using classes may be due to the structural additions. However, the improved performance introduced by semantically meaningful CL classification is a lot more significant.

4.4 Comparison with other works

This section presents a comparison of our work with other works on the PPA task. In order to obtain a fair comparison we have tested our system on the complete data set, including the preposition `of` (cf. Sec. 2). The results are compared with a maximum-entropy method (Ratnaparkhi et al., 1994), transformation-based learning (TBL, Brill and Resnik (1994)), an instantiation of the back-off estimation (Collins and Brooks, 1995) and a memory-based method (Zavrel et al., 1997). All these works have used the same train and test data set. Table 5 presents the comparison.

In all cases, the quoted figures are the best results obtained by the authors; with the exception of the Brill and Resnik (1994) result, which was obtained by Zavrel et al. (1997) using the same method. Originally, TBL was evaluated by Brill and Resnik (1994)

Baseline	lemma	+CL+FF	WN5+FF	WN10+FF	WN15+FF
58.1	77.4	79.1	78.8	78.1	77.9

Table 3: **Learning results for combinations of FF and other sources:** The four leftmost columns indicate the classes added to our basic feature set, `lemma`.

	lemma	lemma+CL	CL200	CL126	CL-PERM
Accuracy	77.4	78.6	77.6	77.6	77.9

Table 4: **Random Classes:** Results with various randomizations strategies.

on a smaller data set.

Although all systems have used the same data, they have not used similar feature sets. Both Collins and Brooks (1995) and Zavrel et al. (1997) have enhanced the feature generation in various ways; as described in this paper, this was also done for *SNOW*.

5 Conclusion

Over several decades, research on high level inferences such as natural language understanding has emphasized programmed systems, as opposed to those that learn. However, experience in AI research over the past few decades shows that it is unlikely that hand programming or any form of knowledge engineering will generate a robust, non-brittle reasoning system in a complex domain.

An approach that puts learning at the center of high level inferencing (Khardon and Roth, 1997; Valiant, 1995) should suggest ways to make progress in massive knowledge acquisition and, in particular, ways of incorporating incomplete and noisy knowledge from various information sources such as different modalities, teachers or experts, into a highly scalable learning process.

The present work made preliminary steps in this direction. We have studied ways to incorporate external knowledge sources into a learning algorithm in order to improve its performance. This investigation was done within the *SNOW* architecture, a sparse network of threshold gates utilizing the Winnow on-line learning algorithm. The linguistic knowledge sources, noun-class datasets, were compiled for general reasons, irrespective of the task studied here. Knowledge incorporation resulted in a statistically significant performance improvement on PPA, a challenging natural language disambiguation task which has been investigated extensively.

Using random noun classes, we have demonstrated that the semantic nature of the external knowledge is essential. In addition, the granularity of the data was shown to play an important role in the learning performance. A highly granular synset classification failed to improve the results.

A lot of future work is to be done in order to substantiate the results presented here, study more tasks and prepare and investigate the effectiveness of other information sources.

References

A. Blum. 1992. Learning boolean functions in an infinite attribute space. *Machine Learning*, 9(4):373–386, October.

E. Brill and P. Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proc. of COLING*.

P. Buitelaar. 1998. *CoreLex: Systematic Polysemy and Underspecification*. Ph.D. thesis, Computer Science Department, Brandeis University, Feb.

M. Collins and J Brooks. 1995. Prepositional phrase attachment through a backed-off modle. In *Proceedings of Third the Workshop on Very Large Corpora*.

I. Dagan, Y. Karov, and D. Roth. 1997. Mistake-driven learning in text categorization. In *EMNLP-97, The Second Conference on Empirical Methods in Natural Language Processing*, pages 55–63, August.

G. DeJong and R. Mooney. 1986. Explanation-based learning: An alternative view. *Machine Learning*, 1(2):145–176.

G. DeJong. 1981. Generalization based on explanations. In *IJCAI*, pages 67–70.

A. R. Golding and D. Roth. 1996. Applying winnow to context-sensitive spelling correction. In *Machine Learning*, pages 182–190.

M. Herbster and M. Warmuth. 1995. Tracking the best expert. In *Proc. 12th International Conference on Machine Learning*, pages 286–294. Morgan Kaufmann.

M. Junker. 1997. Sigir poster: The effectiveness of using thesauri in ir. In *Proc. of International Conference on Research and Development in Information Retrieval, SIGIR*.

Y. Karov and S. Edelman. 1996. Learning similarity-based word sense disambiguation from sparse data. In *Fourth workshop on very large corpora*, pages 42–55, August.

R. Khardon and D. Roth. 1997. Learning to reason. *Journal of the ACM*, 44(5):697–725, Sept. Earlier version appeared in AAAI-94.

J. Kivinen and M. K. Warmuth. 1995. Exponentiated gradient versus gradient descent for linear predictors. In *Proc. of STOC*. Tech Report UCSC-CRL-94-16.

Ratnaparkhi et al. (1994)	Brill and Resnik (1994)	Collins and Brooks (1995)	Zavrel et al. (1997)	*SNOW*
81.6	81.9	84.5	84.4	84.8

Table 5: **System comparison:** Comparison of *SNOW* results with those of previous works. All the quoted figures are the best results obtained by the authors, with the exception of the Brill and Resnik (1994) result which was obtained by Zavrel et al. (1997).

A. Kosmynin and I Davidson. 1996. Using background contextual knowledge for documents representation. In *PODP Workshop*, Palo-Alto.

R. Krovetz and W. B. Croft. 1992. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2):115–141.

N. Littlestone and M. K. Warmuth. 1994. The weighted majority algorithm. *Information and Computation*, 108(2):212–261.

N. Littlestone. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318.

N. Littlestone. 1991. Redundant noisy attributes, attribute errors, and linear threshold learning using Winnow. In *Proc. 4th Annu. Workshop on Comput. Learning Theory*, pages 147–156, San Mateo, CA. Morgan Kaufmann.

M. P. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, June.

George A. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

T.M. Mitchell, R.M. Keller, and S.T. Kedar-Cabelli. 1986. Explanation Based Learning. *Machine Learning*, 1(1):47–80.

A. Ratnaparkhi, J. Reynar, and S. Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *ARPA*, Plainsboro, NJ, March.

P. Resnik. 1992. Wordnet and distributional analysis: A class-based approach to lexical discovery. In *AAAI Workshop on Statistically-based Natural Language Processing Techniques*, pages 54–64, July.

P. Resnik. 1995. Disambiguating noun groupings with respect to wordnet senses. In *Proceedings of the Third Annual Workshop on Very Large Corpora.*

D. Roth and D. Zelenko. 1998. Part of speech tagging using a network of linear separators. In *COLING-ACL 98, The 17th International Conference on Computational Linguistics.*

L. G. Valiant. 1994. *Circuits of the Mind.* Oxford University Press, November.

L. G. Valiant. 1995. Rationality. In *Workshop on Computational Learning Theory*, pages 3–14, July.

J. Zavrel, W. Daelemans, and J. Veenstra. 1997. Resolving pp attachment ambiguities with memory based learning. In *Computational Natural Language Learning*, Madrid, Spain, July.