Proceedings of the Eleventh National Conference on Artificial Intelligence, 1993, AAAI Press / MIT Press, pages 798–803.

A Case-Based Approach to Knowledge Acquisition for Domain-Specific Sentence Analysis

Claire Cardie
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
E-mail: cardie@cs.umass.edu

Abstract

This paper describes a case-based approach to knowledge acquisition for natural language systems that simultaneously learns part of speech, word sense, and concept activation knowledge for all open class words in a corpus. The parser begins with a lexicon of function words and creates a case base of context-sensitive word definitions during a human-supervised training phase. Then, given an unknown word and the context in which it occurs, the parser retrieves definitions from the case base to infer the word's syntactic and semantic features. By encoding context as part of a definition, the meaning of a word can change dynamically in response to surrounding phrases without the need for explicit lexical disambiguation heuristics. Moreover, the approach acquires all three classes of knowledge using the same case representation and requires relatively little training and no hand-coded knowledge acquisition heuristics. We evaluate it in experiments that explore two of many practical applications of the technique and conclude that the case-based method provides a promising approach to automated dictionary construction and knowledge acquisition for sentence analysis in limited domains. In addition, we present a novel case retrieval algorithm that uses decision trees to improve the performance of a k-nearest neighbor similarity metric.

Introduction

In recent years, there have been an increasing number of natural language systems that successfully perform domain-specific text summarization (see [MUC-3 Proceedings 1991; MUC-4 Proceedings 1992]). However, many of the best-performing systems rely on knowledge-based parsing techniques that are extremely tedious and time-consuming to port to new domains. We estimate, for example, that the domain-dependent knowledge engineering effort for the UMass/MUC-3[1] system spanned 1500 person-hours [Lehnert et al. 1991b]. Although the exact type and form of the domain-specific knowledge required by a parser varies from system to system, all knowledge-based language processing systems rely on at least the following information: for each word encountered in a text, the system must (1) know which *parts of speech, word senses, and concepts* are plausible in the given domain and (2) determine which part of speech, word sense, and concepts apply, *given the particular context in which the word occurs*.

Consider, for example, the following sentences from the MUC domain of Latin American terrorism:

1. The terrorists **killed** *General Bustillo*.
2. The *general* concern was that children might be **killed**.
3. In *general*, terrorist activity is confined to the cities.

It is clear that in this domain the word "general" has at least two plausible parts of speech (noun and adjective) and two word senses (a military officer and a universal entity). A sentence analyzer has to know that these options exist and then choose the noun/military officer form of "general" for sentence 1, the adjective/universal entity form in 2, and the noun/universal entity form in 3.

In addition to part of speech and word sense ambiguity, these sentences also illustrate a form of concept ambiguity with respect to the domain of terrorism. Sentence 1, for example, clearly describes a terrorist act — the word "killed" implies that a murder took place and the perpetrators of the crime were "terrorists." This is not the case for sentence 2 — the verb "killed" appears, but no murder has yet occurred and there is no implication of terrorist activity. This distinction is important in the MUC domain where the goal is to extract from texts only information concerning 8 classes of terrorist events including murders, bombings, attacks, and kidnappings. All other information should be effectively ignored. To be successful in this *selective concept extraction* task [Lehnert et al. 1991a], a sentence analyzer not only needs access to word-concept pairings (e.g., the word "killed" is linked to the "terrorist murder" concept), but must also accurately distinguish legitimate concept activation contexts from bogus ones (e.g., the phrase "terrorists killed" implies that a "terrorist murder" occurred, but "children might be killed" probably doesn't).

This paper describes a case-based method for knowledge acquisition that begins with a lexicon of only closed class words and learns the part of speech, general and specific word senses, and concept activation information for

[1] The domain for the MUC-3 and MUC-4 performance evaluations was Latin American terrorism. The general task for each system was to summarize all terrorist events mentioned in a set of 100 previously unseen texts.

all open class words in a corpus.[2] We first create a case base of context-sensitive word definitions during a human-supervised training phase. After training, given an open class word and the context in which it occurs, the parser retrieves the most similar cases from the case base and then uses them to infer syntactic and semantic information for the open class word. No explicit lexical disambiguation heuristics are used, but because context is encoded as part of each definition, the same word may be assigned a different part of speech, word sense, or concept activation in different contexts.

The paper also describes the results of two experiments that explore different, but related applications of this knowledge acquisition technique. In the first application, we assume the existence of a nearly complete domain-specific dictionary and use the case base to infer the features of unknown words. In the second, more ambitious application, we assume only a small dictionary of function words and use the case base to determine the definition of *all* open class words. Although these tasks have been addressed separately in related research, our approach is the first to simultaneously accommodate both using a single mechanism.

Moreover, previous approaches to automated lexical acquisition can be classified along three dimensions: (1) the type of knowledge acquired by the approach, (2) the amount of training data required by the approach, and (3) the amount of knowledge required by the approach. [Brent 1990; Grefenstette 1992; Resnik 1992; and Zernik 1991], for example, present systems that learn either syntactic or limited semantic knowledge but not both. Statistically-based methods that acquire (usually syntactic) lexical knowledge have been successful (e.g., [Brent 1991; Church & Hanks 1990; Hindle 1990; Resnik 1992; Yarowsky 1992; and Zernik 1991]), but these require the existence of very large, often hand-tagged corpora. Finally, there exist knowledge-intensive methods that acquire syntactic and/or semantic lexical knowledge, but rely heavily on hand-coded world knowledge (e.g., [Berwick 1983; Granger 1977; Hastings et al. 1991; Lytinen & Roberts 1989; and Selfridge 1986]) or hand-coded heuristics that describe how and when to acquire new word definitions (e.g., [Jacobs & Zernik 1988 and Wilensky 1991]).

Our approach to knowledge acquisition for natural language systems differs from existing work in its:

- **unified approach to learning lexical knowledge.** The same case-based method and case representation are used to simultaneously learn both syntactic and semantic information for unknown words.
- **encoding of context as part of a word definition.** This allows the definition of a word to change dynamically in response to surrounding phrases and obviates the need for explicit, hand-coded lexical disambiguation heuristics.
- **need for relatively little training.** In the experiments described below, we obtained promising results after training on only 108 sentences. This implies that the method may work well for small corpora where statistical approaches fail due to lack of data.
- **lack of hand-coded heuristics to drive the acquisition process.** These are implicitly encoded in the case base.
- **leveraging of two existing machine learning paradigms.** For case retrieval, we use a decision tree algorithm to improve the performance of a simple k-nearest neighbor similarity metric.

In the remainder of the paper we describe the details of the approach including the case representation, case base construction, and the hybrid approach to case retrieval. We also discuss the results of the two experiments mentioned briefly above.

Case Representation

As discussed in the last section, our goal is to learn part of speech, word sense, and concept activation knowledge for any open class word in a corpus by drawing from a case base of domain-specific, context-sensitive word definitions. However, the case representation relies on three predefined taxonomies, one for each class of knowledge that we're trying to learn. This section, therefore, first briefly describes the taxonomies and then shows how they are used in conjunction with parser-generated knowledge to construct the word definition cases.

The Taxonomies

To start, we set up a taxonomy of allowable word senses. Naturally, these reflect the goals of a particular domain. For the remainder of the paper, we will use the TIPSTER JV corpus as our sample domain. This corpus currently contains over 1300 texts that recount world-wide activity in the area of joint ventures/tie-ups between businesses. A portion of the word sense taxonomy created for the TIPSTER JV domain is shown in Figure 1. The complete taxonomy includes 14 general word senses and 42 specific word senses. They are used to describe all non-verb open class words.

General word sense *Specific word sense*	*Description*
jv-entity	party involved in a tie-up
company-name	name of a company
generic-company-name	e.g. "Co." in "Plastics Co."
government	government-affiliated entity
person	an individual
industry	type of business or industry
research	research and development
production	manufacturing, production
sales	sales, marketing,trade
facility	physical facilities
communications	broadcasting stations
factory	manufacturing sites
farm	agricultural sites
location	location expression
country	country name
city	city name
entity	generic entity

Figure 1: Word Sense Taxonomy (partial)

[2]Closed class words are function words like prepositions, auxiliaries, and connectives, whose meanings vary little from one domain to another. All other words (e.g., nouns, verbs, adjectives) are open class words.

Concept Types	*Description*
tie-up	indicates a tie-up activity
tie-up-secondary	weak indicator of a tie-up
total-capitalization	total cash capitalization
ownership-%	indicates a share in the tie-up
industry	indicates the type of industry
ind-research	performed within the scope
ind-production	of the tie-up

Figure 2: Taxonomy of Concept Types (partial)

Next, we define a taxonomy of 11 domain-specific concept types which represent a subset of the relevant information to be included in the summary of each joint venture text (see Figure 2). Finally, we use a taxonomy of 18 parts of speech (not shown). The taxonomy specifies 7 parts of speech generally associated with open class words and reserves the remaining 11 parts of speech for closed class words. Although the word sense and concept taxonomies are clearly domain-specific, the part of speech taxonomy is parser-dependent rather than domain-dependent. We emphasize, however, that our approach depends not on the specifics of any of the taxonomies, only on their existence.

Representation of Cases

Each case in the case base represents the definition of a single open class word as well as the context in which it occurs in the corpus. It is a list of 39 attribute-value pairs that can be grouped into three sets of features:

- **word definition features (6)** that represent semantic and syntactic knowledge associated with the open class word in the current context
- **local context features (20)** that represent semantic and syntactic knowledge for the two words preceding and the two words following the current word
- **global context features (13)** that represent the current state of the parser

Figure 3 shows the case for the word "venture" in a sentence taken directly from the TIPSTER JV corpus. Examine first the word definition features. The open class **word** defined by this case is "venture" and its part of speech in the current context is a *noun modifier (nm)*.[3] The **gen-ws** and **spec-ws** features refer to the word's general and specific word senses. In this example, "venture" has been assigned the most general word sense, *entity*, and has no specific word senses. The **concept** feature indicates that "venture" activates the domain-specific *tie-up* concept in this context. There is also a **morphol** feature associated with the current word that indicates its class of suffix. The *nil* value used here means that no morphology information was derived for "venture."

Next, examine the local context features. For each of the two words that precede and follow the current open class word (referred to in Figure 3 as **prev1**, **prev2**, **fol1**, and **fol2**), we draw from the taxonomies to specify its part of speech, word senses, and activated concepts. The word immediately following "venture," for example, is the *noun* "firm." It has been assigned the *jv-entity* general word sense because it refers to a business, but has no specific word senses and activates no domain-specific concept in this context.

Finally, examine the global context features that encode information about the state of the parser at the word "venture." When the parser reaches the word "venture," it has recognized two major constituents — the subject and verb phrase. Neither activates any domain-specific concepts, but the subject does have general and specific word senses. These are acquired by taking the union of the senses of each word in the noun phrase. (Verbs are currently assigned no general or specific word senses.) Because the direct object has not yet been recognized, all of its corresponding features in the case are empty. In addition to specifying information about each of the main constituents, the global context features also include syntactic and semantic knowledge for the most recent low-level constituent (**last constit**). A low-level constituent can be either a noun phrase, verb, or prepositional phrase and sometimes coincides with one of the major constituents — the subject, verb phrase, or direct object. This is the case in Figure 3 where the low-level constituent preceding "venture" is just the verb.

Case Base Construction

Using the case representation described in the last section, we create a case base of context-dependent word definitions from a small subset of the sentences in the TIPSTER JV corpus. Because the goal of the approach is to learn syntactic and semantic information for only open class words, we assume the existence of a function word lexicon. This lexicon maintains the part of speech and word senses (if any apply) for 129 function words. None of the function words has any associated domain-specific concepts.

The semi-automated training phase alternately consults a human supervisor and a parser (i.e., the CIRCUS parser [Lehnert 1990]) to create a case for each open class word in the training sentences. More specifically, whenever an open class word is encountered, CIRCUS creates a case for the word, automatically filling in the global context features, the **word** and **morphol** features for the unknown word, and the local context features for the preceding two words (i.e., the **prev1** and **prev2** features). Local context features for the following two words (i.e., **fol1** and **fol2**) will be added to the case after CIRCUS reaches them in its left-to-right traversal of the sentence. The user is consulted via a menu-driven interface only to specify the current word's part of speech, word senses, and concept activation information. These values are stored in the **p-o-s**, **gen-ws**, **spec-ws**, and **concept** word definition features and are used by the parser to process the current word. When CIRCUS finishes its analysis of the training sentences, it has generated one case for every occurrence of an open class word.

[3] The *noun modifier(nm)* category covers both adjectives and nouns that act as modifiers. We reserve the *noun* category for head nouns only.

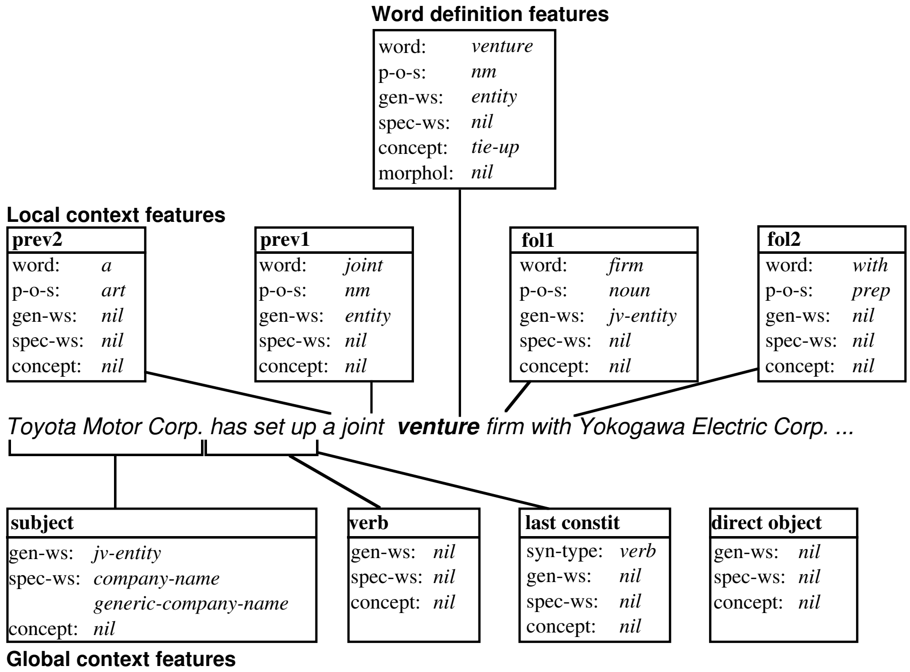

Figure 3: Case for "venture"

Case Retrieval

Once the case base has been constructed, we can use it to determine the definition of new words in the corpus. Assume, for example, that we want to know the part of speech, word senses, and activated concepts for "Toyo's" in the sentence:

> Yasui said this is **Toyo's** and JAL's third hotel joint venture.

First, CIRCUS parses the sentence and creates a probe case for "Toyo's" filling in the **word** and **morphol** features of the case as well as its global and local context features using the method described in the last section.[4] The only difference between a test case and a training case is the **gen-ws**, **spec-ws**, **p-o-s**, and **concept** features for the unknown word. During training, the human supervisor specifies values for these *missing features*, but during testing they are omitted from the case entirely. It is the job of the case retrieval algorithm to find the training cases that are most similar to the probe and use them to predict values for the missing features of the unknown word. We use the following algorithm for this task:

1. Compare the probe to each case in the case base, counting the number of features that match (i.e., match = 1, mismatch = 0). Do not include the missing features in the comparison. Only give partial credit (.5) for matches on *nil*'s.
2. Keep the 10 highest-scoring cases.
3. Of these, return the case(s) whose **word** matches the unknown word, if any exist. Otherwise, return all 10 cases.[5]
4. Let the retrieved cases vote on the values for the probe's missing features.

The case retrieval algorithm is essentially a k-nearest neighbors matching algorithm (k = 10) with a bias toward cases whose **word** matches the unknown word. An interesting feature of the algorithm is that it allows a word to take on a meaning different from any it received during the training phase. However, one problem with the retrieval mechanism is that it assumes that all features are equally important for learning part of speech, word sense, and concept activation knowledge. Intuitively, it seems that accurate prediction of each class of missing information may rely on very different subsets of the feature set. Unfortunately, it is difficult to know which combinations of features will best predict each class of knowledge without trying all of them.

There are machine learning algorithms, like decision tree algorithms (see [Quinlan 1986]), however, that can be used to perform the feature specification task. Very briefly, decision tree algorithms learn to classify objects into one of *n* classes by finding the features that are most important for the classification and creating a tree that branches on each of them until a classification can be made. We use Quinlan's C4.5 decision tree system [Quinlan 1992] to select the features to be included for k-nearest neighbor case retrieval:

1. Given the cases from the training sentences as input, let C4.5 create a decision tree for each missing feature.[6]

---

[4]There is a bootstrapping problem in that the **fol1** and **fol2** features are needed to specify the probe case for "Toyo's." This problem will be addressed in the second experiment. For now, assume that the parser has access to all **fol1** and **fol2** features at the position of the unknown word.

[5]More than 10 cases will be returned if there are ties.

[6]We omit the **p-o-s**, **gen-ws**, **spec-ws**, and **concept** word definition features from training cases because those are the features

2. Note the features that occurred in each tree. This essentially produces, for each of the 4 missing attributes, a list of all features that C4.5 found useful for predicting its value.
3. Instead of invoking the case retrieval algorithm once for each test case, run it 4 times, once for each missing attribute to be predicted. In the retrieval for attribute *x*, however, include only the features C4.5 found to be important for predicting *x* in the k-nearest neighbors calculations.

By using C4.5 for feature specification, we automatically tune the case retrieval algorithm for independent prediction of part of speech, word senses, and concept activation.[7]

Experiment 1

In this section we describe an application that uses the case-based approach described above to determine the definition of unknown words given a nearly complete domain-specific dictionary. We assume the existence of the function word lexicon briefly described above (129 entries) and then create a case base of context-sensitive word definitions for all open class words in 120 sentences from the TIPSTER JV corpus. In each of 10 experiments, we remove from the case base (of 2056 instances) all cases associated with 12 randomly chosen sentences and use these as a test set.[8] For each test case, we then invoke the case retrieval algorithm to predict the part of speech, general and specific word senses, and concept activation information of its unknown word while leaving the rest of the case intact. This experimental design simulates a nearly complete dictionary in that it assumes perfect knowledge of the global and local context of the unknown word.

Figure 4 shows the average percentage correct for prediction of each feature across the 10 runs and compares them

Missing Feature	Case-Based Approach	Random Selection	Default
p-o-s	93.0%	34.3%	81.5%
gen-ws	78.0%	17.0%	25.6%
spec-ws	80.4%	37.3%	58.1%
concept	95.1%	84.2%	91.7%

Figure 4: Experiment 1 Results (% correct for prediction of each feature)

to two baselines.[9] The first baseline indicates the expected accuracy of a system that randomly guesses a legal value for each missing feature based on the distribution of values across the test set. The second baseline shows the performance of a system that always chooses the most frequent value as a default. The default for the concept activation feature (*nil*) achieves quite good results, for example. (This is because relatively few words actually activate concepts in this domain.) Chi-square significance tests on the associated frequencies show that the case-based approach does significantly better than both of the baselines (p = .01).

Experiment 2

In the second application, we assume only a very sparse dictionary (129 function words) and use the case-based approach to acquire definitions of *all* open class words. We use the same experimental design as experiment 1 — we create a case base from 120 TIPSTER JV sentences (2056 cases) and use 10-fold cross validation. During testing, however, we now make no assumptions about the availability of definitions for words surrounding the unknown word. CIRCUS parses each test sentence and creates a test case each time an open class word is encountered, filling in the global context features, the **word** and **morphol** features for the unknown word, and the local context features for the preceding two words. If the following two words are both function words, then **fol1** and **fol2** features can also easily be specified. In most cases, however, one or both of **fol1** and **fol2** are open class words for which the system has no definition. In these cases, the parser makes an educated guess based on the training instances:

1. If the word did not appear during training, fill in the **word** features, but use *nil* as the value for the remaining **fol1** and **fol2** attributes.
2. If the word appeared during training, let each **fol1** and **fol2** feature be the union of the values that occurred in the training phase definitions.

We also relax the k-nearest neighbors matching algorithm and allow a non-empty intersection on any **fol1** or **fol2** feature to count as a full match. (Matches on *nil* still receive only half credit.) Results for experiment 2 are shown in Figure 5 along with the same baseline comparisons from experiment 1. Not surprisingly, all of the results have dropped somewhat; however, chi-square analysis still shows that the performance of the case-based approach is significantly better than the baselines (p =.01).

Missing Feature	Case-Based Approach	Random Selection	Default
p-o-s	91.0%	34.3%	81.5%
gen-ws	65.3%	17.0%	25.6%
spec-ws	74.0%	37.3%	58.1%
concept	94.3%	84.2%	91.7%

Figure 5: Experiment 2 Results (% correct for prediction of each feature)

whose values the decision trees are trying to predict. In addition, we omit the **word**, **prev1-word**, **prev2-word**, **fol1-word**, and **fol2-word** features because of their large branching factor. These "word" features are always included in the k-nearest neighbors calculations, however.

[7] Space limitations preclude the inclusion of experiments that compare the original case retrieval algorithm with the modified version. Those results are discussed in [Cardie 1993], however, which focuses on the contributions of this research to machine learning.

[8] In each experiment, a different set of 12 sentences is chosen. This amounts to a 10-fold cross validation testing scheme.

[9] Note that all results indicate performance for only the open class words. When function words are included, all percentages increase. For part of speech prediction, for example, the case-based results increase from 93.0% to 96.4%.

Conclusions

We have presented a new, case-based approach to the acquisition of lexical knowledge that simultaneously learns 3 classes of knowledge using the same case representation and requires no hand-coded acquisition heuristics and relatively little training. We create a case base of context-sensitive word definitions and use it to learn part of speech, word sense, and concept activation knowledge for unknown words. The case-based technique employs a decision tree algorithm to specify the features relevant for simple k-nearest neighbor case retrieval and allows the definition of a word to change in response to new contexts without the use of lexical disambiguation heuristics. We have tested our approach in two practical applications and found it to perform significantly better than baselines that randomly guess or choose default values for the features of the unknown word. Given results in previous work (see [Cardie 1992]), however, we believe performance can be much improved through the use of case adaptation heuristics that exploit knowledge implicit in the taxonomies that is unavailable to the learning algorithms. In addition, although this paper discusses only two applications of the approach, many more exist. Explicit domain-specific lexicons can be constructed, for example, by saving the definitions acquired during the testing phase of the experiments discussed above. Finally, we have demonstrated that the case-based technique described here is a promising approach to dictionary construction and knowledge acquisition for sentence analysis in limited domains.

Acknowledgments

I wish to thank Professor J. Ross Quinlan for supplying the C4.5 decision tree system. This research was supported by the Office of Naval Research Contract N00014-92-J-1427 and NSF Grant No. EEC-9209623, State/Industry/University Cooperative Research on Intelligent Information Retrieval.

References

Berwick, R. 1983. Learning word meanings from examples. *Proceedings, Eighth International Joint Conference on Artificial Intelligence*. Karlsruhe, Germany, pp. 459-461.

Brent, M. 1991. Automatic acquisition of subcategorization frames from untagged text. *Proceedings, 29th Annual Meeting of the Association for Computational Linguistics*. University of California, Berkeley, Association for Computational Linguistics, pp. 209-214.

Brent, M. 1990. Semantic classification of verbs from their syntactic contexts: automated lexicography with implications for child language acquisition. *Proceedings, Twelfth Annual Conference of the Cognitive Science Society*. Cambridge, MA, The Cognitive Science Society, pp. 428-437.

Cardie, C. 1993. Using Decision Trees to Improve Case-Based Learning. To appear in, P. Utgoff (Ed.), *Proceedings, Tenth International Conference on Machine Learning*. University of Massachusetts, Amherst, MA.

Cardie, C. 1992. Learning to Disambiguate Relative Pronouns. *Proceedings, Tenth National Conference on Artificial Intelligence*. San Jose, CA, AAAI Press/MIT Press, pp. 38-43.

Church, K., & Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16.

Granger, R. 1977. Foulup: A program that figures out meanings of words from context. *Proceedings, Fifth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, pp. 172-178.

Grefenstette, G. 1992. SEXTANT: Exploring unexplored contexts for semantic extraction from syntactic analysis. *Proceedings, 30th Annual Meeting of the Association for Computational Linguistics*. University of Delaware, Newark, DE, Association for Computational Linguistics, pp. 324-326.

Hastings, P., Lytinen, S., & Lindsay, R. 1991. Learning Words from Context. *Proceedings, Eighth International Conference on Machine Learning*. Northwestern University, Chicago, IL.

Hindle, D. 1990. Noun classification from predicate-argument structures. *Proceedings, 28th Annual Meeting of the Association for Computational Linguistics*. University of Pittsburgh, Association for Computational Linguistics, pp. 268-275.

Jacobs, P., & Zernik, U. 1988. Acquiring Lexical Knowledge from Text: A Case Study. *Proceedings, Seventh National Conference on Artificial Intelligence*. St. Paul, MN, Morgan Kaufmann, pp. 739-744.

Lehnert, W. 1990. Symbolic/Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds. In J. Barnden, & J. Pollack (Eds.), *Advances in Connectionist and Neural Computation Theory*. Norwood, NJ, Ablex Publishers, pp. 135-164.

Lehnert, W., Cardie, C., Fisher, D., Riloff, E., & Williams, R. 1991a. University of Massachusetts: Description of the CIRCUS System as Used for MUC-3. *Proceedings, Third Message Understanding Conference (MUC-3)*. San Diego, CA, Morgan Kaufmann, pp. 223-233.

Lehnert, W., Cardie, C., Fisher, D., Riloff, E., & Williams, R. 1991b. University of Massachusetts: MUC-3 Test Results and Analysis. *Proceedings, Third Message Understanding Conference (MUC-3)*. San Diego, CA, Morgan Kaufmann, pp. 116-119.

Lytinen, S., & Roberts, S. 1989. Lexical Acquisition as a By-Product of Natural Language Processing. *Proceedings, IJCAI-89 Workshop on Lexical Acquisition*.

Proceedings, Fourth Message Understanding Conference (MUC-4). 1992. McLean, VA, Morgan Kaufmann.

Proceedings, Third Message Understanding Conference (MUC-3). 1991. San Diego, CA, Morgan Kaufmann.

Quinlan, J. R. 1992. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning*, 1, pp. 81-106.

Resnik, P. 1992. A class-based approach to lexical discovery. *Proceedings, 30th Annual Meeting of the Association for Computational Linguistics*. University of Delaware, Newark, DE, Association for Computational Linguistics, pp. 327-329.

Selfridge, M. 1986. A computer model of child language learning. *Artificial Intelligence*, 29, pp. 171-216.

Wilensky, R. 1991. Extending the Lexicon by Exploiting Subregularities. *Tech. Report No. UCB/CSD 91/618*. Computer Science Division (EECS), University of California, Berkeley.

Yarowsky, D. 1992. Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. *Proceedings, COLING-92*.

Zernik, U. 1991. Train1 vs. Train 2: Tagging Word Senses in Corpus. In U. Zernik (Ed.), *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*. Hillsdale, NJ, Lawrence Erlbaum Associates, pp. 91-112.