In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, 1997.*

A Corpus-Based Approach for Building Semantic Lexicons

Ellen Riloff and Jessica Shepherd
Department of Computer Science
University of Utah
Salt Lake City, UT 84112
riloff@cs.utah.edu

Abstract

Semantic knowledge can be a great asset to natural language processing systems, but it is usually hand-coded for each application. Although some semantic information is available in general-purpose knowledge bases such as WordNet and Cyc, many applications require domain-specific lexicons that represent words and categories for a particular topic. In this paper, we present a corpus-based method that can be used to build semantic lexicons for specific categories. The input to the system is a small set of seed words for a category and a representative text corpus. The output is a ranked list of words that are associated with the category. A user then reviews the top-ranked words and decides which ones should be entered in the semantic lexicon. In experiments with five categories, users typically found about 60 words per category in 10-15 minutes to build a core semantic lexicon.

1 Introduction

Semantic information can be helpful in almost all aspects of natural language understanding, including word sense disambiguation, selectional restrictions, attachment decisions, and discourse processing. Semantic knowledge can add a great deal of power and accuracy to natural language processing systems. But semantic information is difficult to obtain. In most cases, semantic knowledge is encoded manually for each application.

There have been a few large-scale efforts to create broad semantic knowledge bases, such as WordNet (Miller, 1990) and Cyc (Lenat, Prakash, and Shepherd, 1986). While these efforts may be useful for some applications, we believe that they will never fully satisfy the need for semantic knowledge. Many domains are characterized by their own sublanguage containing terms and jargon specific to the field. Representing all sublanguages in a single knowledge base would be nearly impossible. Furthermore, domain-specific semantic lexicons are useful for minimizing ambiguity problems. Within the context of a restricted domain, many polysemous words have a strong preference for one word sense, so knowing the most probable word sense in a domain can strongly constrain the ambiguity.

We have been experimenting with a corpus-based method for building semantic lexicons semi-automatically. Our system uses a text corpus and a small set of seed words for a category to identify other words that also belong to the category. The algorithm uses simple statistics and a bootstrapping mechanism to generate a ranked list of potential category words. A human then reviews the top words and selects the best ones for the dictionary. Our approach is geared toward fast semantic lexicon construction: given a handful of seed words for a category and a representative text corpus, one can build a semantic lexicon for a category in just a few minutes.

In the first section, we describe the statistical bootstrapping algorithm for identifying candidate category words and ranking them. Next, we describe experimental results for five categories. Finally, we discuss our experiences with additional categories and seed word lists, and summarize our results.

2 Generating a Semantic Lexicon

Our work is based on the observation that category members are often surrounded by other category members in text, for example in conjunctions (*lions and tigers and bears*), lists (*lions, tigers, bears...*), appositives (*the stallion, a white Arabian*), and nominal compounds (*Arabian stallion; tuna fish*). Given a few category members, we wondered whether it

would be possible to collect surrounding contexts and use statistics to identify other words that also belong to the category. Our approach was motivated by Yarowsky's word sense disambiguation algorithm (Yarowsky, 1992) and the notion of statistical salience, although our system uses somewhat different statistical measures and techniques.

We begin with a small set of seed words for a category. We experimented with different numbers of seed words, but were surprised to find that only 5 seed words per category worked quite well. As an example, the seed word lists used in our experiments are shown below.

Energy:	*fuel gas gasoline oil power*
Financial:	*bank banking currency dollar money*
Military:	*army commander infantry soldier troop*
Vehicle:	*airplane car jeep plane truck*
Weapon:	*bomb dynamite explosives gun rifle*

Figure 1: Initial Seed Word Lists

The input to our system is a text corpus and an initial set of seed words for each category. Ideally, the text corpus should contain many references to the category. Our approach is designed for domain-specific text processing, so the text corpus should be a representative sample of texts for the domain and the categories should be semantic classes associated with the domain. Given a text corpus and an initial seed word list for a category C, the algorithm for building a semantic lexicon is as follows:

1. We identify all sentences in the text corpus that contain one of the seed words. Each sentence is given to our parser, which segments the sentence into simple noun phrases, verb phrases, and prepositional phrases. For our purposes, we do not need any higher level parse structures.

2. We collect small context windows surrounding each occurrence of a seed word as a head noun in the corpus. Restricting the seed words to be head nouns ensures that the seed word is the main concept of the noun phrase. Also, this reduces the chance of finding different word senses of the seed word (though multiple noun word senses may still be a problem). We use a very narrow context window consisting of only two words, the first noun to the word's right and the first noun to its left. We collected only nouns under the assumption that most, if not all, true category members would be nouns.[1] The context windows do not cut across sentence boundaries. Note that our context window is much narrower than those used by other researchers (Yarowsky, 1992). We experimented with larger window sizes and found that the narrow windows more consistently included words related to the target category.

3. Given the context windows for a category, we compute a category score for each word, which is essentially the conditional probability that the word appears in a category context. The category score of a word W for category C is defined as:

$$Score(W,C) = \frac{freq.\ of\ W\ in\ C's\ context\ windows}{freq.\ of\ W\ in\ corpus}$$

Note that this is not exactly a conditional probability because a single word occurrence can belong to more than one context window. For example, consider the sentence: *I bought an AK-47 gun and an M-16 rifle.* The word *M-16* would be in the context windows for both *gun* and *rifle* even though there was just one occurrence of it in the sentence. Consequently, the category score for a word can be greater than 1.

4. Next, we remove stopwords, numbers, and any words with a corpus frequency ≤ 5. We used a stopword list containing about 30 general nouns, mostly pronouns (e.g., *I, he, she, they*) and determiners (e.g., *this, that, those*). The stopwords and numbers are not specific to any category and are common across many domains, so we felt it was safe to remove them. The remaining nouns are sorted by category score and ranked so that the nouns most strongly associated with the category appear at the top.

5. The top five nouns that are not already seed words are added to the seed word list dynamically. We then go back to Step 1 and repeat the process. This bootstrapping mechanism dynamically grows the seed word list so that each iteration produces a larger category context. In our experiments, the top five nouns were added automatically without any human intervention, but this sometimes allows non-category words to dilute the growing seed word list. A few inappropriate words are not likely to have much impact, but many inappropriate words or a few highly frequent words can weaken the feedback process. One could have a person verify that each word belongs to the target category before adding it to the seed word list, but this

[1] Of course, this may depend on the target categories.

would require human interaction at each iteration of the feedback cycle. We decided to see how well the technique could work without this additional human interaction, but the potential benefits of human feedback still need to be investigated.

After several iterations, the seed word list typically contains many relevant category words. But more importantly, the ranked list contains many additional category words, especially near the top. The number of iterations can make a big difference in the quality of the ranked list. Since new seed words are generated dynamically without manual review, the quality of the ranked list can deteriorate rapidly when too many non-category words become seed words. In our experiments, we found that about eight iterations usually worked well.

The output of the system is the ranked list of nouns after the final iteration. The seed word list is thrown away. Note that the original seed words were already known to be category members, and the new seed words are already in the ranked list because that is how they were selected.[2]

Finally, a user must review the ranked list and identify the words that are true category members. How one defines a "true" category member is subjective and may depend on the specific application, so we leave this exercise to a person. Typically, the words near the top of the ranked list are highly associated with the category but the density of category words decreases as one proceeds down the list. The user may scan down the list until a sufficient number of category words is found, or as long as time permits. The words selected by the user are added to a permanent semantic lexicon with the appropriate category label.

Our goal is to allow a user to build a semantic lexicon for one or more categories using only a small set of known category members as seed words and a text corpus. The output is a ranked list of potential category words that a user can review to create a semantic lexicon quickly. The success of this approach depends on the quality of the ranked list, especially the density of category members near the top. In the next section, we describe experiments to evaluate our system.

[2]It is possible that a word may be near the top of the ranked list during one iteration (and subsequently become a seed word) but become buried at the bottom of the ranked list during later iterations. However, we have not observed this to be a problem so far.

3 Experimental Results

We performed experiments with five categories to evaluate the effectiveness and generality of our approach: *energy*, *financial*, *military*, *vehicles*, and *weapons*. The MUC-4 development corpus (1700 texts) was used as the text corpus (MUC-4 Proceedings, 1992). We chose these five categories because they represented relatively different semantic classes, they were prevalent in the MUC-4 corpus, and they seemed to be useful categories.

For each category, we began with the seed word lists shown in Figure 1. We ran the bootstrapping algorithm for eight iterations, adding five new words to the seed word list after each cycle. After the final iteration, we had ranked lists of potential category words for each of the five categories. The top 45 words[3] from each ranked list are shown in Figure 2.

While the ranked lists are far from perfect, one can see that there are many category members near the top of each list. It is also apparent that a few additional heuristics could be used to remove many of the extraneous words. For example, our number processor failed to remove numbers with commas (e.g., *2,000*). And the military category contains several ordinal numbers (e.g., *10th 3rd 1st*) that could be easily identified and removed . But the key question is whether the ranked list contains many true category members. Since this is a subjective question, we set up an experiment involving human judges.

For each category, we selected the top 200 words from its ranked list and presented them to a user. We presented the words in random order so that the user had no idea how our system had ranked the words. This was done to minimize contextual effects (e.g., seeing five category members in a row might make someone more inclined to judge the next word as relevant). Each category was judged by two people independently.[4]

The judges were asked to rate each word on a scale from 1 to 5 indicating how strongly it was associated with the category. Since category judgements can be highly subjective, we gave them guidelines to help establish uniform criteria. The instructions that were given to the judges are shown in Figure 3.

We asked the judges to rate the words on a scale from 1 to 5 because different degrees of category membership might be acceptable for different applications. Some applications might require strict cat-

[3]Note that some of these words are not nouns, such as *boarded* and *U.S.-made*. Our parser tags unknown words as nouns, so sometimes unknown words are mistakenly selected for context windows.

[4]The judges were members of our research group but not the authors.

Energy: Limon-Covenas[a] oligarchs spill staples poles Limon Barrancabermeja Covenas 200,000 barrels oil Bucaramanga pipeline prices electric pipelines towers Cano substation transmission rates pylons pole infrastructure transfer gas fuel sale lines companies power tower price gasoline industries insurance Arauca stretch inc industry forum nationalization supply electricity controls

---

Financial: monetary fund nationalization attractive circulation suit gold branches manager bank advice invested banks bomb_explosion investment invest announcements content managers insurance dollar savings product employee accounts goods currency reserves amounts money shops farmers maintenance Itagui economics companies foundation moderation promotion annually cooperatives empire loans industry possession

---

Military: infantry 10th 3rd 1st brigade technician 2d 3d moran 6th 4th Gaspar 5th 9th Amilcar regiment sound 13th Pineda brigades Anaya division Leonel contra anniversary ranks Uzcategui brilliant Aristides escort dispatched 8th Tablada employee skirmish puppet Rolando columns (FMLN) deserter troops Nicolas Aureliano Montes Fuentes

---

Vehicle: C-47 license A-37 crewmen plate plates crash push tank pickup Cessna aircraft cargo passenger boarded Boeing_727 luxury Avianca dynamite_sticks hostile passengers accident sons airplane light plane flight U.S.-made weaponry truck airplanes gunships fighter carrier apartment schedule flights observer tanks planes La_Aurora[b] fly helicopters helicopter pole

---

Weapon: fragmentation sticks cartridge AK-47 M-16 carbines AR-15 movie clips knapsacks calibers TNT rifles cartridges theater 9-mm 40,000 quantities grenades machineguns dynamite kg ammunition revolvers FAL rifle clothing boots materials submachineguns M-60 pistols pistol M-79 quantity assault powder fuse grenade caliber squad mortars explosives gun 2,000

[a] Limon-Covenas refers to an oil pipeline.
[b] La_Aurora refers to an airport.

Figure 2: The top-ranked words for each category

CRITERIA: On a scale of 0 to 5, rate each word's strength of association with the given category using the following criteria. We'll use the category ANIMAL as an example.

5: CORE MEMBER OF THE CATEGORY:
If a word is clearly a member of the category, then it deserves a 5. For example, dogs and sparrows are members of the ANIMAL category.

4: SUBPART OF MEMBER OF THE CATEGORY:
If a word refers to a part of something that is a member of the category, then it deserves a 4. For example, feathers and tails are parts of ANIMALS.

3: STRONGLY ASSOCIATED WITH THE CATEGORY:
If a word refers to something that is strongly associated with members of the category, but is not actually a member of the category itself, then it deserves a 3. For example, zoos and nests are strongly associated with ANIMALS.

2: WEAKLY ASSOCIATED WITH THE CATEGORY:
If a word refers to something that can be associated with members of the category, but is also associated with many other types of things, then it deserves a 2. For example, bowls and parks are weakly associated with ANIMALS.

1: NO ASSOCIATION WITH THE CATEGORY:
If a word has virtually no association with the category, then it deserves a 1. For example, tables and moons have virtually no association with ANIMALS.

0: UNKNOWN WORD:
If you do not know what a word means, then it should be labeled with a 0.

IMPORTANT! Many words have several distinct meanings. For example, the word "horse" can refer to an animal, a piece of gymnastics equipment, or it can mean to fool around (e.g., "Don't horse around!"). If a word has ANY meaning associated with the given category, then only consider that meaning when assigning numbers. For example, the word "horse" would be a 5 because one of its meanings refers to an ANIMAL.

Figure 3: Instructions to human judges

egory membership, for example only words like *gun*, *rifle*, and *bomb* should be labeled as weapons. But from a practical perspective, subparts of category members might also be acceptable. For example, if a *cartridge* or *trigger* is mentioned in the context of an event, then one can infer that a gun was used. And for some applications, any word that is strongly associated with a category might be useful to include in the semantic lexicon. For example, words like *ammunition* or *bullets* are highly suggestive of a weapon. In the UMass/MUC-4 information extraction system (Lehnert et al., 1992), the words *ammunition* and *bullets* were defined as weapons, mainly for the purpose of selectional restrictions.

The human judges estimated that it took them approximately 10-15 minutes, on average, to judge the 200 words for each category. Since the instructions allowed the users to assign a zero to a word if they did not know what it meant, we manually removed the zeros and assigned ratings that we thought were appropriate. We considered ignoring the zeros, but some of the categories would have been severely impacted. For example, many of the legitimate weapons (e.g., M-16 and AR-15) were not known to the judges. Fortunately, most of the unknown words were proper nouns with relatively unambiguous semantics, so we do not believe that this process compromised the integrity of the experiment.

Finally, we graphed the results from the human judges. We counted the number of words judged as 5's by either judge, the number of words judged as 5's or 4's by either judge, the number of words judged as 5's, 4's, or 3's by either judge, and the number of words judged as either 5's, 4's, 3's, or 2's. We plotted the results after each 20 words, stepping down the ranked list, to see whether the words near the top of the list were more highly associated with the category than words farther down. We also wanted to see whether the number of category words leveled off or whether it continued to grow. The results from this experiment are shown in Figures 4-8.

With the exception of the Energy category, we were able to find 25-45 words that were judged as 4's or 5's for each category. This was our strictest test because only true category members (or subparts of true category members) earned this rating. Although this might not seem like a lot of category words, 25-45 words is enough to produce a reasonable core semantic lexicon. For example, the words judged as 5's for each category are shown in Figure 9.

Figure 9 illustrates an important benefit of the corpus-based approach. By sifting through a large text corpus, the algorithm can find many relevant category words that a user would probably not enter in a semantic lexicon on their own. For example, suppose a user wanted to build a dictionary of Vehicle words. Most people would probably define words such as *car*, *truck*, *plane*, and *automobile*. But it is doubtful that most people would think of words like *gunships*, *fighter*, *carrier*, and *ambulances*. The corpus-based algorithm is especially good at identifying words that are common in the text corpus even though they might not be commonly used in general. As another example, specific types of weapons (e.g., *M-16*, *AR-15*, *M-60*, or *M-79*) might not even be known to most users, but they are abundant in the MUC-4 corpus.

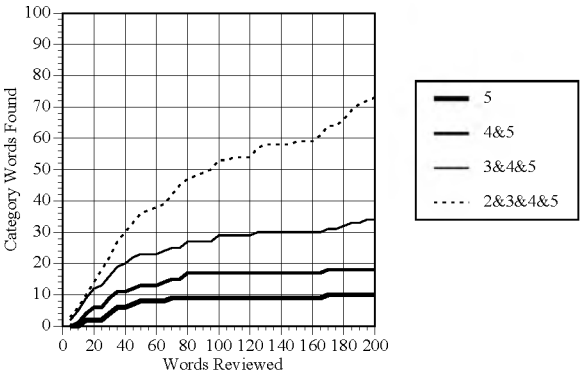

Figure 4: Energy Results

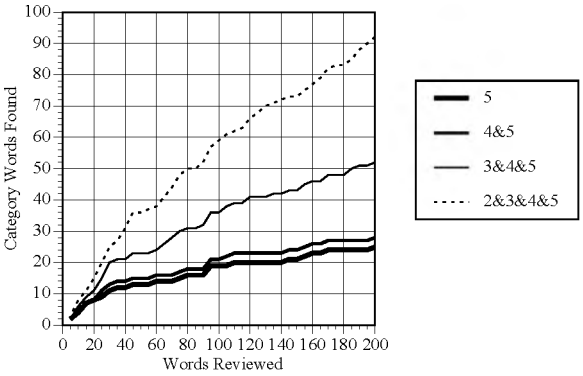

Figure 5: Financial Results

If we consider all the words rated as 3's, 4's, or 5's, then we were able to find about 50-65 words for every category except Energy. Many of these words would be useful in a semantic dictionary for the category. For example, some of the words rated as 3's for the Vehicle category include: *flight*, *flights*, *aviation*, *pilot*, *airport*, and *highways*.

Most of the words rated as 2's are not specific to the target category, but some of them might be useful for certain tasks. For example, some words judged as 2's for the Energy category are: *spill*, *pole*,

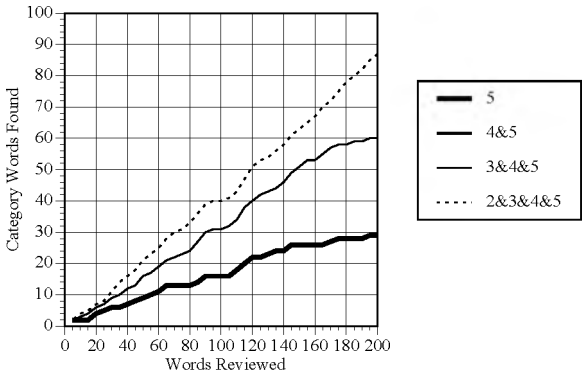

Figure 6: Military Results

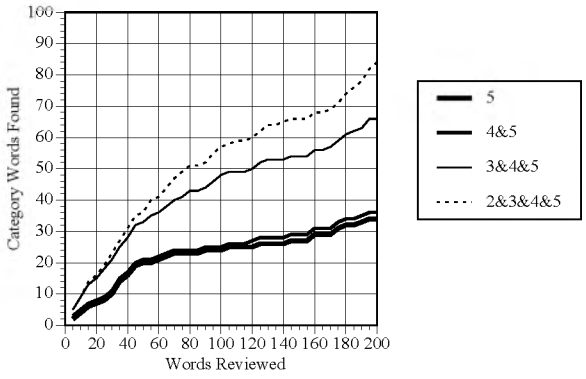

Figure 7: Vehicle Results

tower, and *fields*. These words may appear in many different contexts, but in texts about Energy topics these words are likely to be relevant and probably should be defined in the dictionary. Therefore we expect that a user would likely keep some of these words in the semantic lexicon but would probably be very selective.

Finally, the graphs show that most of the acquisition curves displayed positive slopes even at the end of the 200 words. This implies that more category words would likely have been found if the users had reviewed more than 200 words. The one exception, again, was the Energy category, which we will discuss in the next section. The size of the ranked lists ranged from 442 for the financial category to 919 for the military category, so it would be interesting to know how many category members would have been found if we had given the entire lists to our judges.

4 Selecting Categories and Seed Words

When we first began this work, we were unsure about what types of categories would be amenable to this approach. So we experimented with a number

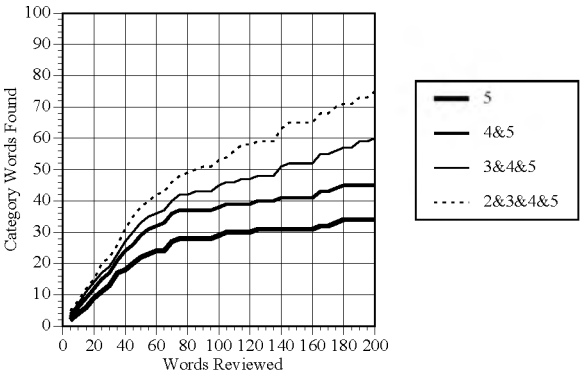

Figure 8: Weapon Results

of different categories. Fortunately, most of them worked fairly well, but some of them did not. We do not claim to understand exactly what types of categories will work well and which ones will not, but our early experiences did shed some light on the strengths and weaknesses of this approach.

In addition to the previous five categories, we also experimented with categories for Location, Commercial, and Person. The Location category performed very well using seed words such as *city*, *town*, and *province*. We didn't formally evaluate this category because most of the category words were proper nouns and we did not expect that our judges would know what they were. But it is worth noting that this category achieved good results, presumably because location names often cluster together in appositives, conjunctions, and nominal compounds.

For the Commercial category, we chose seed words such as *store*, *shop*, and *market*. Only a few new commercial words were identified, such as *hotel* and *restaurant*. In retrospect, we realized that there were probably few words in the MUC-4 corpus that referred to commercial establishments. (The MUC-4 corpus mainly contains reports of terrorist and military events.) The relatively poor performance of the Energy category was probably due to the same problem. If a category is not well-represented in the corpus then it is doomed because inappropriate words become seed words in the early iterations and quickly derail the feedback loop.

The Person category produced mixed results. Some good category words were found, such as *rebel*, *advisers*, *criminal*, and *citizen*. But many of the words referred to organizations (e.g., *FMLN*), groups (e.g., *forces*), and actions (e.g., *attacks*). Some of these words seemed reasonable, but it was hard to draw a line between specific references to people and concepts like organizations and groups that may or may not consist entirely of people. The

Energy: oil electric gas fuel power gasoline electricity petroleum energy CEL

Financial: monetary fund gold bank invested banks investment invest dollar currency money economies loans billion debts millions IMF commerce wealth inflation million market funds dollars debt

Military: infantry brigade regiment brigades division ranks deserter troops commander corporal GN Navy Bracamonte soldier units patrols cavalry detachment officer patrol garrisons army paratroopers Atonal garrison battalion unit militias lieutenant

Vehicle: C-47 A-37 tank pickup Cessna aircraft Boeing_727 airplane plane truck airplanes gunships fighter carrier tanks planes La_Aurora helicopters helicopter automobile jeep car boats trucks motorcycles ambulances train buses ships cars bus ship vehicle vehicles

Weapon: AK-47 M-16 carbines AR-15 TNT rifles 9-mm grenades machineguns dynamite revolvers rifle submachineguns M-60 pistols pistol M-79 grenade mortars gun mortar submachinegun cannon RPG-7 firearms guns bomb machinegun weapons car_bombs car_bomb artillery tanks arms

Figure 9: Words judged as 5's for each category

large proportion of action words also diluted the list. More experiments are needed to better understand whether this category is inherently difficult or whether a more carefully chosen set of seed words would improve performance.

More experiments are also needed to evaluate different seed word lists. The algorithm is clearly sensitive to the initial seed words, but the degree of sensitivity is unknown. For the five categories reported in this paper, we arbitrarily chose a few words that were central members of the category. Our initial seed words worked well enough that we did not experiment with them very much. But we did perform a few experiments varying the number of seed words. In general, we found that additional seed words tend to improve performance, but the results were not substantially different using five seed words or using ten. Of course, there is also a law of diminishing returns: using a seed word list containing 60 category words is almost like creating a semantic lexicon for the category by hand!

5 Conclusions

Building semantic lexicons will always be a subjective process, and the quality of a semantic lexicon is highly dependent on the task for which it will be used. But there is no question that semantic knowledge is essential for many problems in natural language processing. Most of the time semantic knowledge is defined manually for the target application, but several techniques have been developed for generating semantic knowledge automatically. Some systems learn the meanings of unknown words using expectations derived from other word definitions in the surrounding context (e.g., (Granger, 1977; Carbonell, 1979; Jacobs and Zernik, 1988; Hastings and Lytinen, 1994)). Other approaches use example or case-based methods to match unknown word contexts against previously seen word contexts (e.g., (Berwick, 1989; Cardie, 1993)). Our task orientation is a bit different because we are trying to construct a semantic lexicon for a target category, instead of classifying unknown or polysemous words in context.

To our knowledge, our system is the first one aimed at building semantic lexicons from raw text without using any additional semantic knowledge. The only lexical knowledge used by our parser is a part-of-speech dictionary for syntactic processing. Although we used a hand-crafted part-of-speech dictionary for these experiments, statistical and corpus-based taggers are readily available (e.g., (Brill, 1994; Church, 1989; Weischedel et al., 1993)).

Our corpus-based approach is designed to support fast semantic lexicon construction. A user only needs to supply a representative text corpus and a small set of seed words for each target category. Our experiments suggest that a core semantic lexicon can be built for each category with only 10-15 minutes of human interaction. While more work needs to be done to refine this procedure and characterize the types of categories it can handle, we believe that this is a promising approach for corpus-based semantic knowledge acquisition.

6 Acknowledgments

This research was funded by NSF grant IRI-9509820 and the University of Utah Research Committee. We would like to thank David Bean, Jeff Lorenzen, and Kiri Wagstaff for their help in judging our category lists.

References

Berwick, Robert C. 1989. Learning Word Meanings from Examples. In *Semantic Structures: Advances in Natural Language Processing*. Lawrence Erlbaum Associates, chapter 3, pages 89–124.

Brill, E. 1994. Some Advances in Rule-based Part of Speech Tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 722–727. AAAI Press/The MIT Press.

Carbonell, J. G. 1979. Towards a Self-Extending Parser. In *Proceedings of the 17th Annual Meeting of the Association for Computational Linguistics*, pages 3–7.

Cardie, C. 1993. A Case-Based Approach to Knowledge Acquisition for Domain-Specific Sentence Analysis. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 798–803. AAAI Press/The MIT Press.

Church, K. 1989. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the Second Conference on Applied Natural Language Processing*.

Granger, R. H. 1977. FOUL-UP: A Program that Figures Out Meanings of Words from Context. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 172–178.

Hastings, P. and S. Lytinen. 1994. The Ups and Downs of Lexical Acquisition. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 754–759. AAAI Press/The MIT Press.

Jacobs, P. and U. Zernik. 1988. Acquiring Lexical Knowledge from Text: A Case Study. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 739–744.

Lehnert, W., C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. 1992. University of Massachusetts: Description of the CIRCUS System as Used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 282–288, San Mateo, CA. Morgan Kaufmann.

Lenat, D. B., M. Prakash, and M. Shepherd. 1986. CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge-Acquisition Bottlenecks. *AI Magazine*, 6:65–85.

Miller, G. 1990. Wordnet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4).

MUC-4 Proceedings. 1992. *Proceedings of the Fourth Message Understanding Conference (MUC-4)*. Morgan Kaufmann, San Mateo, CA.

Weischedel, R., M. Meteer, R. Schwartz, L. Ramshaw, and J. Palmucci. 1993. Coping with Ambiguity and Unknown Words through Probabilistic Models. *Computational Linguistics*, 19(2):359–382.

Yarowsky, D. 1992. Word sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING-92)*, pages 454–460.