In the Proceedings of the 7th Annual Conference of the University
of Waterloo Centre for the New OED and Text Research,
Oxford, October, 1991.

Noun Homograph Disambiguation Using Local Context in Large Text Corpora

Marti A. Hearst
Computer Science Division, 571 Evans Hall
University of California, Berkeley
Berkeley, CA 94720
and
Xerox Palo Alto Research Center
marti@magnolia.berkeley.edu

Abstract

This paper describes an accurate, relatively inexpensive method for the disambiguation of noun homographs using large text corpora. The algorithm checks the context surrounding the target noun against that of previously observed instances and chooses the sense for which the most evidence is found, where evidence consists of a set of orthographic, syntactic, and lexical features. Because the sense distinctions made are coarse, the disambiguation can be accomplished without the expense of knowledge bases or inference mechanisms. An implementation of the algorithm is described which, starting with a small set of hand-labeled instances, improves its results automatically via unsupervised training. The approach is compared to other attempts at homograph disambiguation using both machine readable dictionaries and unrestricted text and the use of training instances is determined to be a crucial difference.

1 Introduction

Large text corpora and the computational resources to handle them have recently become available to computational linguists. In order to apply to multi-million word corpora, natural language processing techniques must be efficient and domain-independent; for this reason, coarse or partial analyses are becoming more attractive. For example, coarse syntactic interpretation, such as partial parsing (de Marcken 1990)(McDonald 1990) and automatic collocation generation (Smadja & McKeown 1990)(Choueka 1988) are being explored.

This paper describes an accurate, relatively inexpensive method for the disambiguation of English noun homographs, using a large corpus of unrestricted text. (For the purposes of this paper, a homograph is one of two or more words spelled alike but different in meaning.) In the spirit of coarse-grain analysis, the senses distinguished are only the most common, strongly distinct senses of any particular homograph. However, because the interpretation is coarse, the method does not require complex mechanisms used by natural language processing techniques (e.g., as described in (Hirst 1986)), such as knowledge bases, semantic feature representations, or semantic inference.

The algorithm is called CatchWord[1], and its objective is the following: given an English sentence or sentence fragment containing the target noun, determine which of a set of pre-determined senses should be assigned to that noun. This is accomplished by checking the context surrounding the target noun against that of previously observed instances and choosing the sense for which the most evidence is found. The crux of the algorithm lies in the constitution of the evidence for context similarity: a suite of orthographic, syntactic, and lexical features. A training cycle is required for each target noun during which a set of sentences is analyzed, where each sentence is hand-labeled with the correct sense of the noun. The evidence surrounding the target noun is integrated into a characterization of its sense's context. After a sufficient training period, new

[1] or CATCHworD: Corpus-based Automatically Trained Coarse Homograph Disambiguator

unlabeled sentences can be classified based on the characterization that has been acquired. The algorithm can automatically improve its results (using unlabeled sentences) by recording evidence from sentences whose classifications are correct with high certainty.

Although the algorithm requires the overhead of initial training, once the training is complete, a noun's classification is fast. Furthermore, the hand-coding necessary is much simpler than that required for building up the structures associated with more conventional knowledge-based approaches, and investigation is underway for how to do the initial sense-tagging automatically.

Although significant research has been done using machine-readable dictionaries both for acquiring lexical information (e.g., (Markowitz *et al.* 1986), (Calzolari & Bindi 1990), (Chodorow *et al.* 1985), (Alshawi 1987)) and for disambiguation (e.g., (Lesk 1987), (Veronis & Ide 1990), (Jensen & Binot 1987),(Wilks *et al.* 1990)), CatchWord uses unrestricted text corpora based on the assumption that dictionaries do not provide sufficient contextual variety.[2] However, an integration of the results of dictionary-based acquisition with information culled from free-text corpora is preferable to free-text information alone. (These issues are discussed in more detail in Section 6.)

Despite the coarseness of CatchWord's disambiguation capability, there are several applications whose performance potentially will be improved by automated homograph disambiguation. For example, noun-noun compound interpretation, such as that described in (Wu 1990) must both eliminate ambiguities and represent all of the relations in a compound. As a preprocessing step, noun homograph disambiguation could be used to tag the correct sense of, say, *club* in its usage in the noun-noun compound *golf club* (choosing between the stick and the organization sense, both of which are valid for this compound). This should help restrict the number of paths that an interpretation algorithm need explore.

As another example, in an information retrieval task in which relevant sentences are found based on a user query, knowing the sense used in a particular instance would help eliminate inappropriate text fragments. See (Krovetz & Croft 1989) for some preliminary evidence to this effect.

The next two sections describe the motivations behind the Catchword algorithm and the kinds of clues it looks for when performing the sense selection. Next is a detailed description of the algorithm, illustrated with an example sentence, followed by an account of an experiment performed on five noun homographs and a discussion of its results. Next is a discussion of related work and the contributions of some ideas presented here, and the paper concludes with proposals for improving the algorithm and a summary discussion.

2 Motivations

The CatchWord algorithm is motivated by several intuitive observations. First, given a small fragment of text (say, ten words) containing an ambiguous noun, a human reader usually can determine the correct sense in a quick glance. This implies that an algorithm that uses just the local context surrounding the target noun should in most cases be able to correctly determine its sense. This observation is confirmed experimentally for human readers in (Choueka & Luisgnan 1985) and is used by (Kelly & Stone 1975).

Second, because people are exposed to a huge amount of language over the course of their lifetimes, the ability to rapidly recognize the sense of a homograph from its context may stem from a familiarity with these contexts (as opposed to detailed reasoning about the context). An algorithm that gathers similarity information from a large corpus can be thought of as a simulation of the familiarity that arises from frequent exposure.[3] This is related to Becker's claim (Becker 1975) and Wilensky and Aren's demonstration (Wilensky & Arens 1980) that much of language consists of the patchworking together of thousands of short, stereotypical (but often flexible) phrases. Using this idea, a disambiguation algorithm that tries to characterize a sense by its context might be aided by the association of a subset of these phrasal entities with each sense of the target noun. Using phrasal units provides more information than simply associating the sense of the noun with the "content" words that tend to occur near it, and is simpler than attempting to fully interpret the surrounding context.

The final intuition is that either the senses are different enough so that the contexts they tend to inhabit

[2] The terms "unrestricted text" and "free text" refer to text that is not highly structured or restricted to a single type; mainly the terms are used in contrast with machine readable dictionaries.

[3] This is not to say that people's understanding of the context and the meaning of the noun sense does not contribute to their learning what context corresponds to what sense; however, this understanding can be simulated by an algorithm by *identifying* the appropriate context for each sense, whether or not the algorithm actually "understands" the context.

will in most cases have little overlap, or that each of the senses will have features that distinguish it from the others, or both (where features can be syntactic, orthographic, or lexical). If these distinguishing features can be identified and acquired automatically, then even if the noun is found in a partially unfamiliar context (for example, surrounded by content words that have not been previously encountered), the concurrent presence of the distinguishing feature(s) may prove sufficient for sense classification. The complement of this intuition, namely that similar-sense terms will tend to have similar contexts, has been used in statistical processing of large corpora. For example, (Church & Hanks 1990) and (Wilks *et al.* 1990) use frequency of co-occurrence of content words to create clusters of semantically similar words, and (Hindle 1990) uses both simple syntactic frames and frequency of occurrence of content words to determine similarity among nouns.

3 Disambiguation Evidence

In order for CatchWord to succeed, the kinds of evidence it extracts from the training sentences must be selective enough to distinguish among the possible senses, but general enough to cover many possible variations in expression. An approach in which the surrounding context has to be identical to that of a previously encountered one is equivalent to a table lookup, cannot be generalized. At the other end of the spectrum are approaches that consider any occurrence of a lexical item within k words of the target word to be valid evidence.

The criteria used in the experiments reported here fall in between these two extremes. A list of those used in the implemented algorithm appears in Figure 1. Although no semantic *interpretation* is needed, the method *does* make use of a shallow form of semantics: the association between individual lexical items and the different senses of the nouns. This can be considered a semantic property since it is based on the assumption that each sense of the noun will be surrounded by certain topics of discussion, and these topics are, to some extent, indicated by the lexemes used to express them. But more is recorded about the lexical item than simply the fact that it occurs in the same sentence as the target noun: the kind of syntactic construction it occurs in is also registered. The subsequent discussion illustrates some of the observations that motivate the properties listed in Figure 1.

Consider the following sentence fragments, taking note of the nouns *bank* and *tank*:

(1a) `numerous residences,` *banks*`, and libraries`

(1b) `they use holes in trees,` *banks*`, or rocks for nests`

(1c) `are found on the west` *bank* `of the Nile`

(1d) `headed the Chase Manhattan` *Bank* `in New York.`

(2a) `used troops and` *tanks* `to quash a pro democracy`

(2b) `tools and equipment, pumps, hoses, water` *tanks*

(2c) `an enlisted man in the royal air force and` *tank* `corps`

(2d) `case of fighter aircraft the engines and fuel` *tanks*

Even though the fragments are quite small, a reader, without any previous context, is able to determine the correct sense of the noun in each of these cases. What are the characteristics of the data that make this possible?

Sentence (1a) displays a "parallel construction" clue – *banks* is included in a list of nouns, all of which denote institutions that are also buildings. This is true of the "financial" sense of *bank*. Example (1b) also displays a parallel construction, but this time the nouns describe terms that are related according to a more general semantic frame, one that might be called "objects found in nature." This corresponds to the "river bank" sense of *bank* rather than the more specific relation "institutions that are also buildings." In (1c) there are two kinds of pertinent clues: lexical and syntactic. The lexical clues are the river name, *Nile* and the direction, *west*. The syntactic clue is the characteristic pattern *the ["direction"] bank of the "proper name" [river]* (brackets indicate optionality). In (1d) there are also two kinds of clues: lexical (the name of the bank, the location name), and orthographic (the capitalization of the target noun and its modifiers).

Recorded for Target Noun
target is capitalized
target's modifier is capitalized
target is modified
target modifies another item
target is found within a prep phrase headed by one of *in, on, of*
target is found within a prep phrase other than those listed above
a prep phrase adjacent to target is headed by one of *in, on, of*
a prep phrase adjacent to target other than those listed above

Recorded for each Lexical Item
item modifies target
item is modified by target
item is a head in a construct adjacent to target
item is a modifier in a construct adjacent to target
item acts as a verb in an adjacent construct

Figure 1: **Properties used in Disambiguation**

A telling case in which the syntactic frame as well as the associated lexical items must be taken into account is the following: suppose a sentence is known to include both of the items *bank* and *Colorado*. This provides evidence for both of the senses. However, the phrase *... bank of the Colorado ...* can be distinguished from *... Bank of Colorado ...* by two syntactic clues – the lack of the determiner following the *of* preposition in the riverbank sense, and the capitalization of the target word in the institution sense. (CatchWord does not rely solely on these clues: more of the surrounding context is also taken into account.)

In (2a) and (2c), the lexical choices indicate the "military/warfare" semantic frames, thus selecting for the "armored vehicle" sense of *tank*. In (2b) and (2d) are examples of a seemingly strong characteristic of the "container" sense of *tank*: its tendency to be modified. Note however that the lexeme "fuel," although frequently a collocate with *tank* in its "container" sense, also appears frequently near the "vehicle" sense, but in other syntactic frames (e.g. *Plagued by mine fields, a critical shortage of fuel and tanks, and British air superiority ...*).

These illustrations imply that algorithms that rely solely on neighboring content word information will likely make mistakes that could be avoided if simple syntactic information were taken into account. This is not proven conclusively; one could argue that content words suffice for the disambiguation. Even if this assertion is correct, it requires that all potential content words have been previously encountered and associated with the appropriate sense. Since there exist significantly fewer possible (local) syntactic frames than possible neighboring content words (i.e. the algorithm is much more likely to encounter unfamiliar words than unfamiliar syntactic frames), syntactic indicators provide a more generally applicable source of evidence, when combined with lexical co-occurrences, than co-occurrence information alone.

Ideally, the algorithm would make generalizations such as "modified by fluids" for the "container" sense of *tank*, and "associated with action verbs" for the "armored vehicle" sense. As noted in (Choueka 1988), a very large corpus is necessary in order to come across all relevant word combinations, since many lexical items that appear infrequently are nevertheless important as contextual clues. One way to address this problem is to use a lexico-semantic ontology in order to allow generalizations from specific lexical items; for example, if *dinghy* is encountered, and *boat* has been associated with a particular sense in the past, the algorithm should consider attributing to *dinghy* the the evidence associated with *boat*. (Some of the disambiguation techniques based on machine readable dictionaries make generalizations of this sort; this will be elaborated upon below.)

The expectation is that these kinds of properties are enough to characterize and distinguish between the different senses of the nouns. The clues independently do not necessarily determine the noun's sense; whether or not they appear in tandem is also important. However, requiring pairs of clues to occur simultaneously would not only make the evidence-seeking task more expensive, it would also require a larger training set in order to ensure proper coverage. Whether or not this evidence is sufficient, without the addition of inferences based on actual understanding of the text, is a question that should be answered empirically. As will be

seen in Section 4, initial results are quite promising.

4 The CatchWord Algorithm

A general overview of the CatchWord algorithm appeared in the Introduction. This section describes the algorithm in more detail, showing its operation on an example sentence:[4]

```
The female prepares, on the bank of a pond, a nest of water plants.
```

The algorithm operates in one of three modes: supervised training, unsupervised training, and testing mode. A period of supervised training, in which evidence from hand-labeled examples is extracted and recorded, is required as a bootstrap before unsupervised training or testing can begin. The number of training instances required varies depending on the properties of the target noun (see the next section for more detail). The classification of new sentences is done in testing mode. Unsupervised training operates similarly to supervised training, but does not require manual labelling of the sentences; thus unsupervised mode allows for improvement without need for human intervention. Unsupervised mode subsumes testing mode in that new, unlabeled sentences are classified in unsupervised training mode; the difference is that the sentences are used as training instances to improve the algorithm's statistics.

4.1 Preprocessing Stage

A preprocessing stage is required for each mode, and proceeds (automatically) as follows:

1. A sentence or phrase containing an instance of the target noun is isolated from the corpus. (The noun may be functioning as an adjective in a compound, and it may be inflected.)
2. The sentence or phrase is tokenized and tagged with part-of-speech information using a stochastic tagger (Kupiec 1989).
3. The tagged sentence or phrase is segmented into a series of simple prepositional phrases, simple noun phrases, and verb groups. All other information, such as relative pronouns, adverbial phrases, and conjunctions is discarded, and no attempt is made to determine attachment, argument structure, or other larger syntactic relations.[5]

After the preprocessing stage, the training instance consists of an ordered sequence of segments, each labeled with its type (PP, NP, or VG) and consisting of a sequence of tokens tagged with part-of-speech information. The example appears below:

```
(np ((the . det) (female . n))
 vg ((prepares . v3sg))
 pp ((on . prep) (the . det) (bank . n))
 pp ((of . prep) (a . det) (pond . n))
 np ((a . det) (nest . n))
 pp ((of . prep) (water . n) (plants . npl)))
```

4.2 Supervised Training Mode

When in supervised training mode, each sequence is hand-labeled with an integer indicating the appropriate sense of the homograph. Each segment of the sequence is then examined for occurrences of properties of the type appearing in Figure 1 (recall that these properties constitute the evidence for one sense over another). The current version of the algorithm omits several useful kinds of evidence. For example, lists of phrases

[4] Truncated for brevity, from the article entitled "Alligator" in the *American Academic Encyclopedia.*

[5] For the purposes of this task, prepositional phrases are simple noun phrases headed by a preposition, and noun phrases cannot contain prepositions or relative clauses. Modals and auxiliary verbs are also removed from VG's. These simplifications do not seem, in most cases, to adversely affect the processing. However, possibly damaging errors are caused by ignoring conjunctions – for example, adjectival modifiers of noun compounds may be missed (e.g., in the phrase *air and tank power*, the algorithm is not aware that *power* is modified by *air*).

```
The female prepares , on the bank of a pond , a nest of water plants .

  Target vector, index 5, feature SF-TARGET-IN-PP-ON
  #((10 4 21 11 1 0 10 5 9 1 11 10) (11 8 40 3 2 27 2 4 6 2 35 11))
  Target vector, index 10, feature SF-TARGET-ADJ-PP-OF
  #((10 4 21 11 1 0 10 5 9 1 11 10) (11 8 40 3 2 27 2 4 6 2 35 11))
  Term table, index 2, feature SF-TERM-HEAD-IN-ADJACENT-SEGMENT
  term "female" NIL
  term "pond" NIL
  term "nest" NIL
  term "plant" #((0 0 0 0 0) (0 0 0 0 1))
  Term table, index 3, feature SF-TERM-MODIFIES-IN-ADJACENT-SEGMENT
  term "water" #((0 0 0 0 0) (0 0 0 2 0))
  Term table, index 4, feature SF-TERM-VERB-IN-ADJACENT-SEGMENT
  term "prepare" NIL
  Result vector:  #(0.0011765909 0.9988234) Correct Sense:  1
```

Figure 2: Results of Testing Mode after Supervised Training

with parallel structure are not recognized as such (although the adjacency information loosely approximates this phenomenon). Additionally, the presence or absence of definite determiners is not recorded, nor is the relationship between the verb groups and the target noun or other noun phrases. As mentioned above, pairwise evidence indicators are not recorded; e.g., the fact that the "river bank" sense of *bank* often occurs when the target noun is modified by a "direction" word and is followed by a prepositional phrase headed by "of." These kinds of evidence were omitted in order to simplify the algorithm; in future work they and other kinds of evidence will be incorporated.

Two tables of information are maintained for each noun: the target vector and the term table, and evidence is represented by means of frequency counts stored in these tables. The target vector records, for each sense, syntactic and orthographic evidence relating to the target item, such as whether it modifies another item in an NP, or whether it has a capitalized modifier (see the top part of Figure 1).[6] The term table records information relating to the specific lexical items that appear in the segments. Each item is stemmed (using a morphological analyzer (Karttunen *et al.* 1987)) and then stored as evidence supporting the specified sense according to one or more of the classifications in the bottom part of Figure 1.

4.3 Testing Mode

In testing mode, the evidence looked for is identical to that stored during training mode. The classification metric is:[7]

$$\max(CE) \quad where \quad CE_i = \frac{E_i}{\sum_{i=0}^{n-1} E_i} \quad and \quad E_i = \prod_{j=1}^{m}(f_{ij} * 2) + 1$$

CE indicates Comparative Evidence, n = the number of senses, m = the number of evidence features found in the test sentence, and f_{ij} indicates the frequency with which evidence feature j is recorded as occurring with a training sentence containing sense i. The sense with the maximum comparative evidence is chosen for the classification, after an additional check: if the largest CE is not larger than the next largest CE by the amount of a judgement threshold, then the algorithm cannot classify the test sentence. Thus if the evidence is close for the top two senses, no choice is made. There are problems with this metric; this is discussed in Section 7.

The results from testing mode for the sample sentence (after supervised training on 50 examples of each sense of *bank*) appear in Figure 2. Sense 0 is the "financial" sense and sense 1 is the "river edge" sense. The first four lines show evidence associated with the syntactic and orthographic properties of the target noun. The first piece of evidence is that the target appears in a PP segment headed by "on" and this information

[6] Another simplifying assumption is that the final item in an NP or a PP is the head of that phrase; all the rest of the items are considered modifiers.

[7] The author is indebted to John Tukey for suggesting a metric similar to this in order to allow multiplicative combination of the evidence.

is stored in index 5 (indexing is zero-based) in the target vector, which appears on the second line. This evidence has been associated 0 times with sense 0 and 27 times with sense 1. Similarly, the property of being adjacent to a PP headed by "of" occurred 11 times with sense 0 and 35 times with sense 1. The rest of the evidence is associated with lexical items surrounding the target noun. The terms "female," "pond," and "nest" have never been encountered. The term "plant" has, and associated with sense 1, but only when used as a verb. Therefore, it counts evenly as evidence of value 1 for both senses. The term "water" has been encountered as a modifier in an adjacent segment twice before, and so is incorporated into the evidence count. The final normalized vector appears on the last line; there has been more than enough evidence to pass the judgement threshold and classify this sentence as containing sense 1 of *bank*.

4.4 Unsupervised Training Mode

Unsupervised training mode allows evidence to be accumulated without the expense of manual tagging. It operates in a manner similar to that of testing mode, except that if the evidence supporting one of the senses exceeds the combined evidence for all the other senses by a threshold percentage, and if the raw amount of evidence exceeds another parameterized threshold, the sentence is recorded as if it were a manually-tagged training instance.

5 Experiments and Results

A series of experiments was run to test the ideas presented above. Most of the testing was done on sentences extracted from an on-line encyclopedia, the *Academic American Encyclopedia* (Grolier 1990)[8].

One advantage to bootstrapping from an encyclopedia is that a wide range of topics is covered, so there is a likelihood of encountering a noun in a variety of its contexts. Furthermore, if there is an entry or subentry for the target noun, the algorithm is exposed to a dense, context-characteristic discussion of it. On the flip side, a concentrated entry for one sense when there is no corresponding concentration for another can skew the data (this may have happened with *tank*, see results below).

Tests were run on five nouns, three of which were chosen because their main senses' contexts seem distinct intuitively (*bank, tank,* and *bass*), and two because their main senses semm intuitively to be more fuzzy (*country* and *record*). The main senses for each homograph were chosen based on how frequently they appeared in the data; the decision about which sentence displays which sense is the experimenter's judgement, tempered by having looked at dozens of examples. Only the most frequent senses were used in order to ensure an adequate supply of training instances. In detail, the senses are:

tank: The military vehicle vs. the container. The only senses omitted were frozen collocates such as "think tank."

bank: Anything to do with the monetary sense vs. anything to do with the natural object associated with bodies of water. The senses of a row of objects and their collocates such as "data bank" were discarded.

bass: The fish vs. anything to do with the musical sense.

country: The nation sense (including metonymic references to population, government, etc.) vs. the pastoral sense. The latter often appears adjectivally and the meaning is not very distinct from the nation sense.

record: The permanent archive of some event, including both the abstract and the physical notion of the archiving, and including the long series of such events (e.g. *her voting record*) vs. the pinnacle of achievement sense (e.g. *a world record*) vs. the musical disk.

For each training sentence, the sense of its target homograph is labeled by hand and the sentences that are deemed to contain senses outside of those being acquired are removed from the training and testing set (this includes misleading uses of the noun such as sentences mentioning a person whose last name is Banks).

[8] This on-line encyclopedia contains about 32,000 articles and 8.6 million words (64 megabytes). It can be purchased; see the bibliography for the publisher's address.

noun	# trained sentences per sense	# tested sentences per sense	# incorrect sense 0			# incorrect sense 1			% correct both senses
			mc	ut	mu	mc	ut	mu	
tank	20	30	7	0	0	7	0	3	70
	40		2	1	2	7	1	2	75
	60		2	0	0	7	1	1	82
	70		2	0	0	5	0	0	88
bank	20	30	4	1	0	6	0	0	82
	30		4	1	1	4	1	0	82
	40		4	2	0	1	0	2	85
	50		4	0	0	1	1	1	88
bass	10	15	6	1	0	2	0	2	83
	20		0	0	0	2	0	0	93
	25		0	0	0	0	0	0	100
country	20	15	7	1	0	5	0	0	56
	30		4	3	0	3	0	1	63
	40		3	2	0	2	1	0	73
record-1	20	15	3	0	0	0	0	0	90
	30		4	0	0	0	1	0	83
	40		6	0	1	1	0	0	73
record-2	20	15	1	1	0	5	0	1	73
	30		1	1	0	4	1	1	73
	40		1	0	2	3	1	0	77
	mc = misclassified, ut = under threshold (but not mc), mu = both mc and ut								
	judgement threshold = 0.1, max # segments on each side of target = 5								

Table 1: **Results using only Supervised Learning**

noun	# trained per sense superv	# trained per sense unsuperv	# learned correctly		# learned incorrectly		# incorrect (test set)		% correct (test set) both senses
			s0	s1	s0	s1	s0	s1	
tank	20	50	47	47	1	3	0	29	48
	30	40	37	17	1	22	1	19	67
	40	30	27	28	2	1	2	7	85
	50	20	18	18	2	1	3	8	82
	60	10	9	10	1	0	2	4	90
bank	10	40	37	28	3	10	3	9	80
	20	30	30	18	0	12	3	7	83
	30	20	15	15	3	5	5	3	87
	40	10	10	10	0	0	4	2	90
bass	10	15	13	12	2	2	1	2	90
	15	10	9	7	1	2	0	1	97
	20	5	4	4	1	0	0	0	100
country	10	30	16	21	12	8	10	4	53
	20	20	15	14	5	4	9	5	53
	30	10	5	8	3	1	11	2	53
	superv = trained in supervised mode, unsuperv = trained in unsupervised mode								
	Same # training instances and parameters as in Table 1, judgment threshold = 0.1								

Table 2: **Results using both Supervised and Unsupervised Learning**

In all experiments, no overlap is permitted between training and testing sentences, All tests for a particular noun are done on the same set of testing sentences and as the size of the training set increases, all training instances from the previous training set are incorporated into the new training set.

Table 1 presents the results when the algorithm is trained using only supervised mode. Incorrect classifications are divided into three groups: misclassifications not caught by the check against the difference threshold, correct classifications not strong enough to pass the threshold, and misclassifications caught by the threshold check. Technically, the latter are not errors since in these cases the algorithm indicates that it cannot make a decision. Varying the judgement threshold didn't yield any conclusive results, although in some cases it seems that after a good training base has been established, a stricter threshold does more harm than good. This fact may indicate problems with the evaluation metric; ideally the more training received, the more differentiated the classification should be.

All words were stemmed before being recorded, but no stop list was used. Experimentation with putting Grolier's most common words on a stop list (or giving them less weight than other words) did not improve the results significantly.

As can be seen from the table, the homographs whose senses are semantically distinct achieve good results given sufficient training; in the high 80's to 100% correct for identification of *bass's* senses. Some misclassifications are caused, at least in part, by tagging and phrase-identification errors, but usually the culprit is lack of exposure to unusual words. In some cases of misclassification, the term has been associated with the sense, but in a different syntactic configuration than the one in the test sentence. It may seem this differentiation is harmful, but in other cases differentiating by usage helps make the correct distinction (see the *fuel tank* example above). A better solution would be to weight the importance of a particular term based on how frequently it appears in the entire corpus and how frequently it appears with each sense. If it is relatively frequent for both senses, but with a different syntactic usage pattern, it can still be used as evidence. However, if it has no distinguishing pattern, it should not be discarded when found.

The results shown in the tables are for allowing five segments to appear on either side of the target noun. Errors may have been caused by an impoverished context – if the target appears at the end of the sentence, no context to its right is included. This state of affairs is not true to the algorithm but the current software cannot handle multi-sentential input. A few experiments were done allowing three and seven segments to either side of the noun, but these degraded the results. A potential modification to the algorithm is to give it the results of computing over three different sizes of context, or better yet, have it determine when it is necessary to enlarge the context in order to increase the classification confidence.

The homographs with the less well-distinguished senses achieved results in the low 70's, although these results may have been better had more training instances been available. The test labeled "record-1" is an experiment in distinguishing the "archived event" ¿from the "pinnacle achievement" sense. In the "record-2" test, the "archived event" was contrasted with the "musical disk" sense. Surprisingly, in the "record-1" test, performance became worse with more training examples. A test in which all three were pitted against one another yielded very bad results; this may be due to insufficient training instances and/or problems with the classification metric.

Inspection of *country*'s misclassifications reveals some insight. Two of the sentences are polysemous enough that human classification was difficult (e.g., in the fragment *89% of the country is virtually unsettled*, both senses are implied). Several misclassifications are caused by errors in the noun-phrase recognizer, i.e. an inability to handle hyphenated terms obstructs discernment of crucial relationships. For example, *country* is a modifier in the phrase *country-western singer* (in its other sense, *country* almost never modifies another term).

Some preliminary tests were run on the transferability of the results ¿from one text genre (*American Academic*) to another (*New York Times*). The initial results are good: after training the algorithm on 100 of Grolier's *bank* examples, the algorithm successfully classified 27 out of 30 instances of sense 0 and 7 out of 9 instances for sense 1, for a total of 87% correct classifications. To test the results on a completely different genre, all sentences containing the word *bank* from Lewis Carroll's *Alice's Adventures in Wonderland* and *Through the Looking Glass*, were tested. All 8 sentences referred to the "river" sense of *bank* and 6 of these were correctly classified.

Table 3 presents the results from running the algorithm in unsupervised mode after an initial run in supervised mode. The column headed "# learned correctly" indicates how many of the unsupervised training sentences were classified correctly and then recorded as evidence. Similarly, "# learned incorrectly" indicates

how many were classified with the wrong sense. The total number from the "# trained" column minus the sum of these two columns indicates the number of sentences not used because their comparative evidence does not pass the unsupervised training judgement threshold.

Classification of the homographs with distinct senses proved successful, thus showing that the algorithm can improve itself automatically, without the need for hand-labeled sentences. However, a good training base need be already obtained, or else too many erroneous classifications are made, sometimes resulting in very skewed results (see *tank* in Table 2).

It is interesting to note that the results in general are higher than when training is completely supervised. This may happen because the sentences that are discarded (due to not passing the threshold) are problematic ones for the algorithm, and by eliminating them some skew in the data is avoided.

The results with the not-well-distinguished homograph *country* were quite poor. This may indicate some limitations on the capabilities of unsupervised training, although a higher judgement threshold may help improve the results.

6 Relation to Other Work

This section describes other research on homograph disambiguation, but only those approaches that do not rely on detailed semantic encoding. Wherever possible, the experimental results shown above are compared. A discussion of the relationship between other research and that presented here concludes the section.

6.1 Approaches using Large Corpora

An early approach to automated homograph disambiguation based on information culled from a large text corpus is described in (Kelly & Stone 1975). At that time stemming and part-of-speech tagging tools were not available, and so a large amount of their effort was focused on determining which part-of-speech a homograph displayed (categorical disambiguation, to use Hirst's term). Not surprisingly, the results for disambiguation between senses possessing the same part-of-speech were quite low although specific summary numbers are not mentioned.

This algorithm associates with each homograph an ordered set of rules, each consisting of tests to be run on the homograph's local context. The tests include checks for particular lexical items or part-of-speech tags on words at specific "distances" from the target homograph and control information for when to apply the tests. This information was hand-coded by thirty researchers (the project was active for about seven years) who examined numerous examples of the homographs embedded in short concordance lines. CatchWord differs primarily in that: it automatically determines which tests to apply and in what order (equivalently, with what weight); it uses more detailed syntactic information; and it is geared toward and is successful at disambiguating senses possessing the same part-of-speech.

More recent is Zernik's (Zernik 1991) attempt at homograph disambiguation. This approach associates a list of weighted terms, called a signature, with each sentence in the corpus that contains the target homograph, and then hierarchically clusters the signatures (the clusters are post-edited, apparently by hand). Presumably a new instance is matched against the clustered signatures and its position in the cluster indicates its sense (how this is actually done is not explained). The terms in the signature include the probability that the words appear in a window of five words to either side of the target, the probability that a particular word collocates with the target, the probability that the target assumes a particular part of speech, and some microfeatures (not explained in detail). In this work, probability is essentially the frequency of the word in the corpus divided by the number of words in the corpus. Unfortunately, as the author states, this algorithm is unsuccessful at making distinctions between senses possessing the same part-of-speech. (No percentages or other results are provided.)

Hindle and Rooth (Hindle & Rooth 1991) describe a method for determining prepositional phrase attachment using statistical frequency information based on simple syntactic frames (e.g. for *Moscow sent soldiers into Afganistan*, does *into* attach to *sent* or *soldiers*?). The main idea is to record the frequency of occurrence of each preposition with each verb and noun in the corpus. Then when a {noun, verb, preposition} trio occurs and when an assignment cannot be made based on "indisputable" criteria, the attachment is determined by whichever co-occurrence probability is higher (noun-prep or verb-prep).

The term "indisputable" is introduced here to refer to a property of a language that makes automatic, always-correct classification of some phenomenon possible. For example, one of Hindle and Rooth's indisputable observations is that "a preposition is attached to the verb if the noun phrase head is a pronoun."

The results of this algorithm are quite promising: lexical attachment was correct 88.5% and 86.4% of the time for phrases whose underlying relationship is argument-noun and argument-verb, and 84.5% correct overall, for the cases where the confidence in the results was greater than 95%. Like CatchWord, this algorithm makes use of simple syntactic relations but uses no knowledge-base semantics. It has the advantage of not requiring a hand-labeling stage – use of the indisputable cases along with the frequency data is sufficient. The authors state that for the more difficult cases, though, more detailed information will be required.

Another use of indisputable criteria appears in the work of Brent (Brent 1991), in which subcategorization frame information is gathered for verbs from free text corpora. For example to find a verb that takes a subcategorization frame that includes both a direct object and an infinitive, the algorithm finds patterns in the text of the form "V PRONOUN to V" rather than the more error-prone form "V NP to V". In this approach, any uncertain data is disposed of in favor of obtaining high accuracy, so a very large corpus is required in order to get good coverage. The disadvantage of these algorithms is that they appear only to work for phenomena with distinct syntactic structures, and so their sphere of applicability may be limited.

6.2 Approaches using Machine Readable Dictionaries

Recently there have been several attempts to perform lexical disambiguation using information extracted from machine readable dictionaries (MRD's). Lesk's well-known algorithm (Lesk 1987) uses a count of the overlap between the words in the sentence being disambiguated and the words in the dictionary definitions of each of the senses of the words in that sentence to make a classification. He states that his results have accuracies of 50-70% on non-dictionary texts. This number is difficult to compare to CatchWord's results since Lesk's algorithm attempts to classify each ambiguous word according to its corresponding dictionary sense; this is a finer-grained distinction than that made in the experiments described above.

In many cases, the appropriate overlap does not occur. Wilks et al. attempt to account for this problem by extending Lesk's approach by widening the context around each definition (Wilks *et al.* 1990). They do this by computing lexical neighborhood information for all the words in the controlled vocabulary of an entire dictionary (*The Longman's Dictionary of Contemporary English* – LDOCE) via co-occurrence statistics. They then run a clustering algorithm on the neighborhood information in an attempt to partition the words according to the senses they correspond to, and use these partitions to classify the senses. Classification is iterative – first the most local neighborhoods are compared, then neighbors of the neighbors are compared, and so on. They report results for classifying occurrences of the homograph *bank*, but only for its appearance in LDOCE definitions. They try to classify each instance of *bank* into one of the thirteen LDOCE senses, and not surprisingly, get low results (around 53% at best). When they simplify the task to classifying instances into one of the main, coarse senses, they achieve good results, in the 85-90% range. No other homographs were tested in this experiment. These results may be comparable to those described for CatchWord, although neither experiment tests enough words to make a conclusive judgement. Furthermore, if text other than LDOCE definitions had been tested on, the results may have differed.

Guthrie et al. (Guthrie *et al.* 1991) further extend this approach by imposing partitions on the co-occurrence information by making use of the subject code information. (In LDOCE, codes such "Economic" and "Engineering" are associated with some of the sense definitions to indicate a general topic with which the sense is often associated.) This imposes a partition on the co-occurrence network so that words like "account" and "river" are not grouped together in relation to "bank." For homograph disambiguation, again word overlap counts are computed, but in this experiment, words considered to be in the same neighborhood must have the same subject code marking. After the dominant subject code is determined, the algorithm must choose among the senses that share that subject code. The words from the competing senses are then used in another round of iterative overlap count. Disambiguation of two example sentences containing *bank* are described, one of which is: *We got a bank loan to buy a car.* The proper sense is determined to belong either to the Automotive subject area or the Economic subject area. After a total of 38 iterations through word neighborhoods, the correct sense is chosen. Neither "loan" nor "buy" are associated with the primary class of words in the Economic subject area, but after 13 iterations the subject code is chosen on the basis of having both "buy" and "get" present. However, if the sentence had been *We got a bank loan to buy a*

safe car., the Automotive subject area would have been chosen after the initial iteration. Results beyond the two examples are not reported.

Veronis and Ide (Veronis & Ide 1990) declare that the approach of (Wilks *et al.* 1990) suffers from problems similar to Lesk's in that it loses information about how the words in the definitions are interconnected. They convert dictionary definitions into a connectionist network consisting of word nodes for the lexical items and sense nodes for the dictionary senses corresponding to the words. The nodes corresponding to the words in the sentence to be disambiguated are activated, and the senses corresponding to one word mutually inhibit one another. After running a "winner-take-all" strategy, the most active sense for each word is chosen. Due to the connectivity of the network, a context is found automatically. The approach seems promising but no results are reported. The authors note that incorporation of syntactic information would help the algorithm.

6.3 Discussion

From the discussion above it seems that CatchWord's performance is comparable to, or better than other existing approaches, although the dearth of tested results for other approaches makes it impossible to make a definite statement to this effect. However, in the context of these comparisons, some interesting discussion points come to light:

6.3.1 Training Instances vs. Clustering and Indisputable Criteria

The clustering algorithms of Wilks *et al.* and Zernik, subject code partitioning of Guthrie *et al.*, and the connectionist topology of Veronis and Ide can all be seen as attempts to automatically bias[9] the lexical co-occurrence information in order to make it accurately reflect the usage of words corresponding to each homograph sense. This brings to light a crucial aspect of the CatchWord algorithm: the effect of its training phase on correctly labeled data is to delimit the space in which evidence for each sense should reside. These other approaches use secondary information in order to guess the bias, whereas CatchWord gets its information directly from experience with homographs in their "natural habitat," within some text genre.

Thus, Veronis and Ide's approach may benefit from a training period in which groups of connections' weights that are found in association with a particular sense are strengthened. Similarly, adding an initial training cycle (and more thorough contextual information) to the Zernik algorithm may improve its results.

The role of the use of indisputable criteria by algorithms such as Hindle and Rooth's is played in CatchWord's approach by the introduction of training instances. This is necessary since there is no equivalent of general indisputable criteria for homograph disambiguation.

6.3.2 Unrestricted Corpus vs. Dictionary as Data

Lesk mentions that an advantage of his algorithm is that it does not depend on "global" information, i.e., just because one sense A occurs more frequently throughout a corpus than sense B, this does not mean that sense B does not occur. In effect, the choice of which terms appear in a sense's dictionary definition is an implicit indication of meaningful co-occurrence information. Furthermore, this co-occurrence information is among the most prototypical that could be associated with this sense. However, often in free text the most prototypical context is not what surrounds an occurrence of a homograph's sense. CatchWord attempts to account for these situations as well as the more typical ones. In other words, CatchWord is also based on the assumption that the context that is found around an instance of a sense of the homograph is meaningfully related to that sense. But it is reasonable to hypothesize that a larger variety of context is encountered in the free corpus than in the dictionary definitions. (E.g., most dictionaries don't mention that Donald Trump is closely associated with discussions of finance, whereas probably the most salient word associated with this name is "money.") As further evidence, in the experiments of (Guthrie *et al.* 1991), words from the definition's example sentences are included as well as words from the definitions themselves, presumably to add more variety to the neighborhood information.

Furthermore, because the definitions in an MRD present the prototypical or generic uses of words, information extracted from them is not easily adaptable to a variety of text genres. An advantage to an algorithm that extracts information from free text is that it can be automatically trained on and tuned for

[9] The term *bias* is used here in the sense used in Machine Learning as introduced in (Mitchell 1980).

new genres. This is related to the observation that an MRD is inherently a bounded resource whereas an approach using unrestricted text can automatically integrate unfamiliar words, provided that its training base is good.

MRD's word distribution should be more even than that of an unrestricted corpus. It should be noted, however, that within MRD's there are skew effects with respect to semantic frames, which lead to improperly balanced co-occurrence information. For example, as (Wilks *et al.* 1990) note, LDOCE devotes a great deal of discussion to the "robbing" aspect of *bank*, whereas this semantic frame occurs only rarely in the Grolier corpus.

Nevertheless, Catchword would be greatly aided by the word connectivity information available from an MRD in order to make generalizations such as taking the evidence associated with *boat* when *dinghy* is encountered, as discussed above. As an approximation toward that goal, plans are underway to incorporate WordNet, (Miller *et al.* 1990), a large hand-built thesaurus, into the algorithm. Of course, if *boat* has more than one sense some problems will arise from generalizations such as this; empirical experience will determine how much of a problem this is.

7 Proposed Improvements

This section describes some ways the algorithm should be improved. Other candidate improvements were suggested in earlier sections.

7.1 Bootstrapping from Bilingual Corpora

The time required for hand-labeling the training sentences is prohibitive, but there is a way it might be automated. Recently several researchers (e.g. (Brown *et al.* 1991), (Dagan *et al.* 1991), (Inoue 1991)) have suggested using bilingual aligned corpora in the lexical disambiguation task, (the term "aligned" indicates that within the bilingual database, sentences that are translations of one another are grouped together). Although most of the discussion is in terms of choosing words for translating one language to another, it is also suggested that, because the words that have more than one sense in language A have only one sense or a different set of senses in language B, by using a bilingual dictionary, the correct sense of a word in language A can be determined by comparing it with its translation in B. This disambiguation method is of limited applicability, of course, because it requires a bilingual corpus. However, a corpus of translated text could be used to bootstrap CatchWord, providing it with initial training instances (sentences containing a target homograph tagged with its sense), thereby eliminating the hand-labeling step.

7.2 Revisions to the Evidence Metric

The metric itself is currently too naïve. First, it should have a facility to automatically adjust the weights of certain kinds of evidence depending on how important each type is for each noun. Categorizing clustering methods, like that of (Chen & Shrager 1989) may prove useful for this, and (Brown *et al.* 1991) presents an interesting method for ordering decisions of binary choices. Furthermore, the metric is too dependent on frequency information – if it encounters a lot of evidence for one sense and little for another, it becomes unjustifiably biased toward the sense for which many examples have been seen. Related to this is the idea, discussed above, that the importance of a particular term occurring near the target should be adjusted depending on how frequently it appears in the entire corpus, and based on the syntactic patterns it assumes. This indicates a need for some kind of probabilistic technique; (Magerman & Marcus 1991) discuss some ways around the pitfalls in existing probabilistic models.

Additionally, some evidence types should be added to the procedure. For example: what types of adjectives modify head nouns; if a noun is *not* modified by anything other than a determiner; certain combinations of evidence types (as mentioned above, e.g. capitalization plus determiner, lists of parallel constructions, etc); and in what role arguments act with respect to nearby verbs.

Finally, the implementation needs better determination of heads of phrases, and better parses of phrases, especially those containing conjunctions.

7.3 Efficiency Considerations

Once the initial instances have been set up, and parameters have been set, training takes only a few minutes for a particular word and after training is done, classification is extremely fast. If the classification of the training set is automated, then this algorithm will have little overhead in comparison with approaches requiring hand-coded knowledge-bases.

Another consideration is the storage requirements for each homograph. Without the term information these requirements are negligible: a short vector for each sense of the homograph. However a vector for each homograph for each lexical item that is encountered in the text may also be required. This spells potentially unacceptable storage requirements, but this extreme will not be necessary in practice. Over time, it will become clear which terms do not play a role in the classification of the senses and so their entries will be dropped. Furthermore, if a mechanism allowing grouping of like terms is incorporated, redundant information can be made more compact.

8 Conclusion

This paper has shown that noun homograph disambiguation can be accomplished by making use of simple syntactical and lexico-semantic information culled from a large corpus. Its results are comparable to or better than earlier efforts using MRDs and large corpora, but more importantly, it provides a means for the integration of information ¿from MRD's with that of large corpora towards the goal of more robust natural language processing. Crucial to the success of the algorithm is the judicious use of sentences from the corpus to place a bias on the lexical co-occurrence data. Also crucial is the use of partial syntactic information, which is richer than what most statistically-based disambiguation techniques use. This syntactic information is computationally feasible due to the recent development of robust stemmers and part-of-speech taggers.

The next steps will be to integrate data from WordNet into the CatchWord algorithm, to investigate the automation of the hand-labeling of training data by using aligned bilingual corpora, and to make the evaluation metric more robust.

Acknowledgements. Per-Kristian Halvorsen, Peter Norvig, Penni Sibun, John Tukey, Robert Wilensky and Dekai Wu read and provided invaluable comments on earlier drafts of this paper. Marie desJardins' help was crucial during the early stages of this work and Doug Cutting, Jan Pedersen, and Julian Kupiec provided cheerful technical assistance. This work was supported by an internship at Xerox Palo Alto Research Center (and could not have been done without the resources there) and by ONR grant number N00014-89-J-3205.

References

Alshawi, H. (1987). Processing dictionary definitions with phrasal pattern hierarchies. *American Journal of Computational Linguistics*, 13(3):195–202.

Becker, J. (1975). The phrasal lexicon. In R. Schank & B. L. Nash-Webber, editors, *Proceedings of the First Interdisciplinary Workshop on Theoretical Issues in Natural Language Processing*, Cambridge, MA.

Brent, M. R. (1991). Automatic acquisition of subcategorization frames from untagged, free-text corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics.*

Brown, P. F., S. A. D. Pietra, V. J. D. Pietra, & R. L. Mercer (1991). Word sense disambiguation using statistical methods. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 264–270.

Calzolari, N. & R. Bindi (1990). Acquisition of lexical information from a large textual italian corpus. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, Helsinki.

Chen, F. R. & J. Shrager (1989). Automatic discovery of contextual factors describing phonological variation. In *Proceedings of the DARPA Speech and Natural Language Workshop.* Morgan Kaugmann.

Chodorow, M. S., R. Byrd, & G. Hiedorn (1985). Extracting semantic hierarchies from a large on-line dictionary. *Proceedings of the 23th Annual Meeting of the Association for Computational Linguistics*, pages 299–304.

Choueka, Y. (1988). Looking for needles in a haystack, or locating interesting collocational expressions in large textual databases. *Proceedings of the RIAO*, pages 609–623.

Choueka, Y. & S. Luisgnan (1985). Disambiguation by short contexts. *Computers and the Humanities*, 19(3):147–157.

Church, K. & P. Hanks (1990). Word association norms, mutual information, and lexicography. *American Journal of Computational Linguistics*, 16(1):22–29.

Dagan, I., A. Itai, & U. Schwall (1991). Two languages are more informative than one. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 130–137.

de Marcken, C. G. (1990). Parsing the lob corpus. *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 243–252.

Grolier (1990). *Academic American Encyclopedia*. Grolier Electronic Publishing, Danbury, Connecticut.

Guthrie, J. A., L. Guthrie, Y. Wilks, & H. Aidinejad (1991). Subject-dependent co-occurence and word sense disambiguation. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*.

Hindle, D. (1990). Noun classification from predicate-argument structures. *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 268–275.

Hindle, D. & M. Rooth (1991). Structural ambiguity and lexical relations. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*.

Hirst, G. (1986). *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge.

Inoue, N. (1991). Automatic noun classification by using japanese-english word pairs. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 201–208.

Jensen, K. & J.-L. Binot (1987). Disambiguating prepositional phrase attachments by using on-line dictionary definitions. *American Journal of Computational Linguistics*, 13(3):251–260.

Karttunen, L., K. Koskenniemi, & R. M. Kaplan (1987). A compiler for two-level phonological rules. In M. Dalrymple, editor, *Tools for Morphological Analysis*. Center for the Study of Language and Information, Stanford, CA.

Kelly, E. & P. Stone (1975). *Computer recognition of english word senses*, volume 13 of *North-Holland Linguistics Series*. North-Holland, Amsterdam.

Krovetz, R. & B. Croft (1989). Word sense disambiguation using machine-readable dictionaries. In *Proceedings of the Conference on Research and Development in Information Retrieval*, pages 127–136, Cambridge, MA.

Kupiec, J. (1989). Augmenting a hidden markov model for phrase-dependent word tagging. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 92–98. Morgan Kaufmann.

Lesk, M. (1987). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC Conference*, Canada.

Magerman, D. M. & M. P. Marcus (1991). Pearl: A probabilistic chart parser. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 193–199.

Markowitz, J., T. Ahlswede, & M. Evens (1986). Semantically significant patterns in dictionary definitions. *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 112–119.

McDonald, D. D. (1990). Robust partial-parsing through incremental, multi-level processing: rationales and biases. In P. S. Jacobs, editor, *Text-Based Intelligent Systems: Current Research in Text Analysis, Information Extraction, and Retrieval*, pages 61–65. GE Research & Development Center, TR 90CRD198.

Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, & K. J. Miller (1990). Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244.

Mitchell, T. M. (1980). The need for biases in learning generalizations. Technical Report CBM-TR-117, Rutgers University, Department of Computer Science.

Smadja, F. A. & K. R. McKeown (1990). Automatically extracting and representing collocations for language generation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 252–259.

Veronis, J. & N. M. Ide (1990). Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, volume 2, pages 389–394, Helsinki.

Wilensky, R. & Y. Arens (1980). PHRAN - a knowledge based natural language understander. *Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics*, pages 117–121.

Wilks, Y. A., D. C. Fass, C. ming Guo, J. E. McDonald, T. Plate, & B. M. Slator (1990). Providing machine tractable dictionary tools. *Journal of Computers and Translation*, 2.

Wu, D. (1990). Probabilistic unification-based integration of syntactic and semantic preferences for nominal compounds. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, Helsinki.

Zernik, U. (1991). TRAIN1 vs. TRAIN2: Tagging word senses in corpus. In *RIAO 91 Conference Proceedings*, pages 567–585, Barcelona, Spain.