A Method for Disambiguating Word Senses in a Large Corpus

William A. Gale
Kenneth W. Church
David Yarowsky

AT&T Bell Laboratories
600 Mountain Avenue
P. O. Box 636
Murray Hill NJ, 07974-0636

Abstract

Word sense disambiguation has been recognized as a major problem in natural language processing research for over forty years. Both quantitive and qualitative methods have been tried, but much of this work has been stymied by difficulties in acquiring appropriate lexical resources, such as semantic networks and annotated corpora. In particular, much of the work on qualitative methods has had to focus on ''toy'' domains since currently available semantic networks generally lack broad coverage. Similarly, much of the work on quantitative methods has had to depend on small amounts of hand-labeled text for testing and training.

We have achieved considerable progress recently by taking advantage of a new source of testing and training materials. Rather than depending on small amounts of hand-labeled text, we have been making use of relatively large amounts of parallel text, text such as the Canadian Hansards, which are available in multiple languages. The translation can often be used in lieu of hand-labeling. For example, consider the polysemous word *sentence*, which has two major senses: (1) a judicial sentence, and (2), a syntactic sentence. We can collect a number of sense (1) examples by extracting instances that are translated as *peine*, and we can collect a number of sense (2) examples by extracting instances that are translated as *phrase*. In this way, we have been able to acquire a considerable amount of testing and training material for developing and testing our disambiguation algorithms.

The availability of this testing and training material has enabled us to develop quantitative disambiguation methods that achieve 92 percent accuracy in discriminating between two very distinct senses of a noun such as *sentence*. In the training phase, we collect a number of instances of each sense of the polysemous noun. Then in the testing phase, we are given a new instance of the noun, and are asked to assign the instance to one of the senses. We attempt to answer this question by comparing the context of the unknown instance with contexts of known instances using a Bayesian argument that has been applied successfully in related tasks such as author identification and information retrieval.

The Bayesian classifier requires an estimate of $Pr(w|sense)$, the probability of finding the word w in a particular context. Care must be taken in estimating these probabilities since there are so many parameters (e.g., 100,000 for each sense) and so little training material (e.g., 5,000 words for each sense). We have found that it helps to smooth the estimates obtained from the training material with estimates obtained from the entire corpus. The idea is that the training material provides poorly measured estimates, whereas the entire corpus provides less relevant estimates. We seek a trade-off between measurement errors and relevance using a novel interpolation procedure that has one free parameter, an estimate of how much the conditional probabilities $Pr(w|sense)$ will differ from the global probabilities $Pr(w)$. In the sense tagging application, we expect quite large differences, more than 20% of the vocabulary behaves very differently in the conditional context; in other applications such as author identification, we expect much smaller differences and find that less than 2% of the vocabulary depends very much on the author.

The ''sense disambiguation'' problem covers a broad set of issues. Dictionaries, for example, make use of

quite a number of other factors in addition to ‘‘meaning,’’ such as part of speech, etymology, register, dialect and collocations, in deciding when to break an entry into multiple ‘‘senses’’. These other aspects of sense disambiguation may well each require specific methods. In particular, part of speech disambiguation is probably best handled as a separate issue, and indeed, there has been considerable progress on that problem. The proposed method is probably most appropriate for those aspects of sense disambiguation that are closest to the information retrieval task. In particular, the proposed method was designed to disambiguate senses that are usually associated with different topics (e.g., judicial sentences vs. syntactic sentences).

1. Bar-Hillel’s Characterization of the Word-Sense Disambiguation Problem

Word sense disambiguation has been recognized as a major problem in natural language processing research for over forty years. One can find a number of early references, e.g., Kaplan (1950), Yngve (1955), Bar-Hillel (1960), Masterson (1967). Early on, there was a clear awareness that word-sense disambiguation is an important problem to solve:

> ‘‘The basic problem in machine translation is that of multiple meaning,’’ (Masterson, 1967)

But unfortunately, there was also a clear awareness that the problem is very difficult. Bar-Hillel, who had been one of the early leaders in machine translation, abandoned the field when he couldn’t see how a program could disambiguate the word *pen* in the very simple English discourse:

> Little John was looking for his toy box.
> Finally he found it.
> *The box was in the pen.*
> John was very happy.

Bar-Hillel (1960, p. 159) argued that:

> ‘‘Assume, for simplicity’s sake, that *pen* in English has only the following two meanings: (1) a certain writing utensil, (2) an enclosure where small children can play. I now claim that no existing or imaginable program will enable an electronic computer to determine that the word *pen* in the given sentence within the given context has the second of the above meanings, whereas every reader with a sufficient knowledge of English will do this ‘automatically.’ ’’

Bar-Hillel’s real objection was an empirical one. Using his numbers,[1] it appears that programs, at the time, could disambiguate only about 75% of the words in a sentence (e.g., 15 out of 20). Moreover, considering that 8 of the 15 words are unambiguous, the system is really only getting slightly more than half of the polysemous cases. And many of the polysemous cases involve relatively easy issues such as part of speech. It is possible that back in the 1960s, most systems were not much better than chance at disambiguating the kinds of cases that we are concerned with here.

---

1. ‘‘Let me state rather dogmatically that there exists at this moment no method of reducing the polysemy of the, say, twenty words of an average Russian sentence in a scientific article below a remainder of, I would estimate, at least five or six words with multiple English renderings, which would not seriously endanger the quality of the machine output. Many tend to believe that by reducing the number of initially possible renderings of a twenty word Russian sentence from a few tens of thousands (which is the approximate number resulting from the assumption that each of the twenty Russian words has two renderings on the average, while seven or eight of them have only one rendering) to some eighty (which would be the number of renderings on the assumption that sixteen words are uniquely rendered and four have three renderings apiece, forgetting now about all the other aspects such as change of word order, etc.) the main bulk of this kind of work has been achieved, the remainder requiring only some slight additional effort.’’ (Bar-Hillel, 1960, p. 163)

Nevertheless, it should be fairly clear that 75% is really not very good, and is probably inadequate for most applications that one would probably be interested in. Fortunately, it seems to be possible to achieve considerably higher performance these days, and consequently, with 20-20 hindsight, we can afford to be somewhat more optimistic about the prospects of automatic sense disambiguation.

2. Knowledge Acquisition Bottleneck

One of the big problems with the early work on sense-disambiguation is that it did not have a good way of dealing with the knowledge acquisition bottleneck. As Bar-Hillel realized, people have a large set of facts at their disposal, and it is not obvious how a computer could ever hope to gain access to this wealth of knowledge.

> ‘‘Whenever I offered this argument to one of my colleagues working on MT, their first reaction was: ‘But why not envisage a system which will put this knowledge at the disposal of the translation machine?’ Understandable as this reaction is, it is very easy to show its futility. What such a suggestion amounts to, if taken seriously, is the requirement that a translation machine should not only be supplied with a dictionary but also with a universal encyclopedia. This is surely utterly chimerical and hardly deserves any further discussion. Since, however, the idea of a machine with encyclopedic knowledge has popped up also on other occasions, let me add a few words on this topic. The number of facts we human beings know is, in a certain very pregnant sense, infinite.’’

In our view, the crux of the problem is to find a strategy for acquiring this knowledge with a reasonable chance of success. We think that we have found such a strategy by turning to parallel text as a source of testing and training materials. It is likely that parallel texts will become available in ever increasing quantities as more and more translators start to use electronic workstations, and as more and more nations (especially in Europe) adopt policies like the Canadians have, that require vast numbers of documents to be translated into two or more languages.

In fact, just as Bar-Hillel has suggested, much of the work on word-sense disambiguation has been stymied by difficulties in acquiring appropriate lexical resources (e.g., semantic networks, annotated corpora, dictionaries, thesauruses, etc.). Most of the previous work falls into one of three camps:

1. Qualitative Methods, e.g., Hirst (???)
2. Dictionary-based Methods, e.g., Lesk (1986)
3. Corpus-based Methods, e.g., Kelly and Stone (1975)

In each case, the work has been limited by knowledge acquisition bottleneck. For example, there has been a tradition in parts of the AI community of building large experts by hand, e.g., Granger (1977), Rieger (1977), Small and Rieger (198X), Hirst (???). Unfortunately, this approach is not very easy to scale up, as many researchers have observed:

> ‘‘The expert for THROW is currently six pages long, ... but it should be 10 times that size.’’ (Small and Reiger, 198X)

Since this approach is so difficult to scale up, much of the work has had to focus on ‘‘toy’’ domains (e.g., Winograd’s Blocks World) or sublanguages (e.g., Isabelle (1984), Hirschman (1986)). Currently, it is not possible to find a semantic network with the kind of broad coverage that would be required for unrestricted text. If Bar-Hillel were here today, he might argue that it is unlikely that such a broad-coverage semantic network will ever become available, and he might have a point.

From an AI point of view, it appears that the word-sense disambiguation problem is ‘‘AI-Complete,’’ meaning that you can't solve this problem until you've solved all of the other hard problems in AI. Since this is unlikely to happen any time soon (if at all), it would seem to suggest that word-sense disambiguation is just too hard a problem, and we should spend our time working on a simpler problem where we have a good chance of making progress. Rather than accept this rather pessimistic conclusion, we prefer to reject the premise and search for an alternative point of view.

2.1 Machine-Readable Dictionaries (MRDs)

Others such as Lesk (1986), Walker (1987), Ide (1990, Waterloo Meeting) have turned to machine-readable dictionarys (MRD) such as Oxford's Advanced Learner's Dictionary of Current English (OALDCE) in the hope that MRDs might provide a way out of the knowledge acquisition bottleneck. These researchers seek to develop a program that could read an arbitrary text and tag each word in the text with a pointer to a particular sense number in a particular dictionary. Thus, for example, if Lesk's program was given the phrase *pine cone*, it ought to tag *pine* with a pointer to the first sense under *pine* in OALDCE (a kind of evergreen tree), and it ought to tag *cone* with a pointer to the third sense under *cone* in OALDCE (fruit of certain evergreen trees). Lesk's program accomplishes this task by looking for overlaps between the words in the definition and words in the text ‘‘near’’ the ambiguous word.

Unfornatuately, the approach doesn't seem to work as well as one might hope. Lesk (1986) reports accuracies of 50-70% on short samples of *Pride and Prejudice*. Part of the problem may be that dictionary definitions are too short to mention all of the collocations (words that are often found in the context of a particular sense of a polysemous word). In addition, dictionaries have much less coverage than one might have expected. Walker (1987) reports that perhaps half of the words occurring in a new text cannot be related to a dictionary entry.

Thus, like the AI approach, the dictionary-based approach is also limited by the knowledge acquisition bottleneck; dictionaries simply don't record enough of the relevant information, and much of the information that is stored in the dictionary is not in a format that computers can easily digest, at least at present.

2.2 Approaches Based on Hand-Annotated Corpora

A third line of research makes use of hand-annotated corpora. Most of these studies are limited by the availability of hand-annotated text. Since it is unlikely that such text will be available in large quantities for most of the polysemous words in the vocabulary, there are serious questions about how such an approach could be scaled up to handle unrestricted text. Nevertheless, we are extremely sympathetic with the basic approach, and will adopt a very similar strategy ourselves. However, we will introduce one important difference, the use of parallel text in lieu of hand-annotated text.

Kelly and Stone (1975) built 1815 disambiguation models by hand, selecting words with a frequency of at least 20 in a half million word corpus. They started from key word in context (KWIC) concordances for each word, and used these to establish the senses they perceived as useful for content analysis. The models consisted of an ordered set of rules, each giving a sufficient condition for deciding on one classification, or for jumping to another rule in the same model, or for jumping to a rule in the model for another word. The conditions of a given rule could refer to the context within four words of the target word. They could test the morphology of the target word, an exact context word, or the part of speech or semantic class of any of the context words. The sixteen semantic classes were assigned by hand.

Most subsequent work has sought automatic methods because it is quite labor intensive to construct these rules by hand. Weiss (1973) first built rule sets by hand for five words, then developed automatic procedures for building similar rule sets, which he applied to additional three words. Unfortunately, the system was tested on the training set, so it is difficult to know how well it actually worked.

Black (1987, 1988) studied five 4-way polysemous words using about 2000 hand tagged concordance lines for each word. Using 1500 training examples for each word, his program constructed decision trees based on the presence or absence of 81 ‘‘contextual categories’’ within the context[2] of the ambiguous word. He used three different types of contextual categories: (1) subject categories from LDOCE, the Longman Dictionary of Contemporary English (Longman, 1978), (2) the 41 vocabulary items occurring most frequently within two words of the ambiguous word, and (3) the 40 vocabulary items excluding function words occurring most frequently in the concordance line. Black found that the dictionary categories produced the weakest performance (47 percent correct), while the other two were quite close at 72 and 75 percent correct, respectively.[3]

There has recently been a flurry of interest in approaches based on hand-annotated corpora. Hearst (1991) is a very recent example of an approach somewhat like Black (1987, 1988), Weiss (1973) and Kelly and Stone (1975), in this respect, though she makes use of considerably more syntactic information than the others did. Her performance also seems to be somewhat better than the others', though it is difficult to compare performance across systems.

2.3 Two Languages are Better than One

Dagan (1991) argued that ''two languages are better than one'' and showed that it was possible to use the differences between certain languages (Hebrew and German, in his case) in order to obtain certain leverage on word meanings. We have achieved considerable progress recently by following up on this strategy. Rather than depending on small amounts of hand-labeled text, we have been making use of relatively large amounts of parallel text, text such as the Canadian Hansards, which are available in multiple languages. The translation can often be used in lieu of hand-labeling. For example, consider the polysemous word *sentence*, which has two major senses: (1) a judicial sentence, and (2), a syntactic sentence. We can collect a number of sense (1) examples by extracting instances that are translated as *peine*, and we can collect a number of sense (2) examples by extracting instances that are translated as *phrase*. In this way, we have been able to acquire a considerable amount of testing and training material for developing and testing our disambiguation algorithms.

It is important to distinguish the monolingual word-sense disambiguation problem from the translation issue. It is not always necessary to resolve the word-sense ambiguity in order to translate a polysemous word. Especially in related languages like English and French, it is common for word-sense ambiguity to be preserved in both languages. For example, both the English noun *interest* and the French equivalent *interêt* are multiply ambiguous in both languages in more or less the same ways. Thus, one cannot turn to the French to resolve the ambiguity in the English, since the word is equally ambiguous in both languages.

Moreover, when one word does translate to two (e.g., *sentence* → *peine* and *phrase*), the choice of target translation need not indicate a sense split in the source. A famous example of this is the group of Japanese words translated by ''wearing clothes'' in English. While the Japanese have five different words for ''wear'' depending on what part of the body is involved, we doubt that English speakers would ever sort ''wearing shoes'' and ''wearing shirt'' into separate categories.

These examples indicate that word-sense disambiguation and translation are somewhat different problems. It would have been nice if the translation could always be used in lieu of hand-tagging to resolve the word-sense ambiguity but unfortunately, this is not the case. Nevertheless, the translation is often helpful for resolving the ambiguity. It seems to us to make sense to use the translation when it works, and to resort to some alternative such as hand-annotation for the remainder.

---

2. The context was defined to be the concordance line, which we estimate to be about ± 6 words from the ambiguous word, given that his 2000 concordance lines contained about 26,000 words.

3. Although 75% may be less than some performance figures cited earlier, it is possible that Black's figures represent a considerable improvement because a four-way decision is much harder than a two-way decision.

3. An Information Retrieval (IR) Approach to Sense Disambiguation

We have been experimenting with an information retrieval approach to sense disambiguation. In the training phase, we collect a number of instances of *sentence* that are translated as *peine*, and a number of instances of *sentence* uses that are translated as *phrase*. Then in the testing phase, we are given a new instance of *sentence*, and are asked to assign the instance to one of the two senses. We attempt to answer this question by comparing the context of the unknown instance with contexts of known instances.

Basically we are treating contexts as analogous to documents in an information retrieval setting. Just as the probabilistic retrieval model (Salton, 1989, section 10.3) sorts documents *d* by

$$score(d) = \prod_{token \text{ in } d} \frac{Pr(token | rel)}{Pr(token | irrel)}$$

we will sort contexts *c* by

$$score(c) = \prod_{token \text{ in } c} \frac{Pr(token | sense_1)}{Pr(token | sense_2)}$$

where $Pr(token | sense)$ is an estimate of the probability that *token* appears in the context of $sense_1$ or $sense_2$.

Contexts are defined arbitrarily as a window 50 words to the left and 50 words to the right of the ambiguous word. Most previous studies have employed a much narrower notion of context, perhaps only 5 words to the left and 5 words to the right of the ambiguous word, based on the observation that people do not seem to require the extra context. Nevertheless, it is fairly clear that extra context does provide a lot of useful information and since the program is not performing as well as people do, it can use all of the help it can get. For this reason, we have decided to adopt a much broader notion of context than what can be found in most previous studies.

The performance on *sentence* and *drug* is very encouraging (95% and 94%, respectively). We should mention, however, that performance depends very strongly on the choice of words, and that other polysemous words may be more difficult to disambiguate.

These words may be particularly well-suited for this sense-disambiguation procedure because their contexts often contain some very strong clues such as *prices* and *abuse*. Table 1 below shows some words that have a large value for

$$freq(token, sense_1) \log \frac{Pr(token | sense_1)}{Pr(token | sense_2)}$$

These words tend to play an important role in the scoring of contexts. Note that many of the words in Table 1 are, in fact, highly associated with the sense that they are listed under. This suggests that the statistics are doing something fairly reasonable.[4]

4. Accents have been removed from the words in this table.

Table 1: Contextual Clues for Sense Disambiguation		
Word	Sense	Contextual Clues
drug	medicaments	prices, prescription, patent, increase, generic, companies, upon, consumers, higher, price, consumer, multinational, pharmaceutical, costs
drug	drogues	abuse, paraphernalia, illicit, use, trafficking, problem, food, sale, alcohol, shops, crime, cocaine, epidemic, national, narcotic, strategy, head, control, marijuana, welfare, illegal, traffickers, controlled, fight, dogs
sentence	peine	inmate, parole, serving, a, released, prison, mandatory, judge, after, years, who, death, his, murder
sentence	phrase	I, read, second, amended, ‘‘, ’’, protects, version, just, letter, quote, word, ..., last, amendment, insults, assures, quotation, first

4. Six Polysemous Words

We will focus on six polysemous words: *duty, drug, land, language, position* and *sentence*. Table 2 shows the six English nouns, and two French translations for each of the nouns. The final column shows the number of times that each English noun was found with the particular French translation in the corpus.

Table 2: Six Polysemous Words			
English	French	sense	N
duty	droit	tax	1114
	devoir	obligation	691
drug	médicament	medical	2992
	drogue	illicit	855
land	terre	property	1022
	pays	country	386
language	langue	medium	3710
	langage	style	170
position	position	place	5177
	poste	job	577
sentence	peine	judicial	296
	phrase	grammatical	148

We selected these nouns because they could be disambiguated by looking at their French translation in the Canadian Hansards. As mentioned above, the polysemous noun *interest*, for example, would not meet this constraint because the French target *interêt* is just as ambiguous as the English source. In addition, it is important that there be an adequate number of instances of both translations in corpus. It turns out that it is difficult to find words that frequently appear in two or more senses in the Hansards. For instance, the word *bank*, perhaps the canonical example of a polysemous word, turns out not be be very polysemous in this corpus; *bank* is overwhelmingly translated as *banque* (a place for depositing money), and consequently, it is hard to find an adequate number of instances of any of the other senses of *bank*. Part of the problem may be due to the unusual language of the Hansard corpus, which is hardly a balanced sample of general language.[5] Hopefully, we will be able to acquire more diverse sources of parallel text as more and more translations become available in computer-readable form.

5. What Is a Sense?

Although we all share the intuition that words have multiple meanings, it can be difficult to make this intuition precise.[6] One might define the word-sense disambiguation task as Lesk did, and ask the program to tag each word in a novel text with a pointer to a particular sense in a particular dictionary. Rosamund Moon (personal communication), a lexicographer with Oxford University Press, has recently coined the term *flogging* to describe this task, especially when performed by a human rather than by a machine.[7]

Unfortunately, lexicographers do not always agree on how to split a dictionary entry into senses. Fillmore and Atkins (1991), for example, found three senses for the noun *risk*, and observed that most dictionaries failed to list at least one of the three senses. Despite efforts to make the notion of sense more rigorous by a number of linguists (e.g., Quine (1960), Weinreich (1980), Cruse (1986)), it is fairly clear that dictionaries often disagree with one another, as can be seen by comparing a randomly selected pair of dictionaries.

Moreover, some sense distinctions are larger than others. It is not very easy to decide when to lump two senses into one and when to split one sense into two. In some cases, meaning is probably best thought of as a continuous quantity, with an infinite number of ‘‘shades’’ between any two points. The trade-off between lumping and splitting is often fairly arbitrary.

From our perspective, the most serious problem with the flogging task is that there are numerous reasons why a dictionary might split an entry into multiple senses, only some of which have to do with ‘‘meaning’’. Dictionaries may split an entry when there are differences in:

1. part of speech (common),
2. syntactic features such as count/uncount nouns, attributive vs. predicative adjectives, person, number, gender, etc.,
3. valency structures (e.g., transitive vs. intransitive verbs),
4. pronunciation (rare and usually not the only reason for splitting senses),

---

5. In fact, the secondary sense of *bank* (e.g., the edge of a river) is much more common in the Birmingham Corpus, a carefully balanced corpus of general English which was instrumental in the construction of the Cobuild dictionary (Sinclair, 1987). We find that 30% of the instances of *bank* in the Birmingham Corpus refer to the edge of a river, in contrast to our collection of Canadian Hansards, where less than 1% of the instances of *bank* refer to the edge of a river. The language of the Hansards should be regarded as a sublanguage. As such, one would expect to find a smaller vocabulary and less polysemy than one would find in a more balanced sample of general language.

6. It is possible to demonstrate that the intuition behind word-senses has some psycholinguistic validity. Jorgensen (1990), for example, was able to show that subjects could sort words in sentence contexts (drawn from the Brown Corpus (Kucera and Francis, 1967)) into senses. Although her subjects used fewer senses than one would find in a dictionary, she was able to establish that her subjects all share the notion that many words have more than one sense.

7. Rosamund Moon coined the term *flogging* in order to spoof the suggestion that she ought to tag a large corpus in this way by-hand. She had been arguing that one ought to *flag* the words in the context of the ambiguous word with various semantic tags. Not only is the flogging task extremely labor-intensive and tedious (especially when performed by-hand), but it quickly points out a number of inadequacies in the dictionary. Lexicographers tend to know about many of the problems, but it is understandable that they might not want to have them highlighted in such a graphic way. Lexicographers tend to be somewhat more comfortable with the flagging task because it tends to suggest ways to improve the dictionary, without making the flaws quite so obvious. Nevertheless, it appears that Oxford University Press is going ahead with a fairly major *flogging* effort in conjunction with DEC.

5. etymology (rare, especially in learners’ dictionaries; more common in dictionaries based on historical principles),
6. capitalization (e.g., *He* = ‘‘god’’; *East* = ‘‘(formerly) Communist Countries’’; *East* = ‘‘Orient’’),
7. register (e.g., rude, slang, substandard language),
8. dialect (e.g., US, British, Canadian),
9. collocations, phrases: (e.g., *eat away at*, *eat in*, *eat into*, *eat up*), and/or
10. subject codes (subject codes are usually not given in the written text but they can be found in the electronic versions of a few dictionaries, especially CED2 and LDOCE).

In this paper, we would like to focus more on differences in ‘‘meaning’’ and less on differences syntax, etymology, etc. Starting with Kelly and Stone (1975), it has become common practice to unbundle the part of speech issue from the others. Kelly and Stone reported results separately for the two kinds of discriminations, which we interpret as 95% accuracy for part of speech and 77% accuracy for meaning. It makes sense to unbundle the part of speech issue because part of speech is probably best handled with a special purpose tool such the ones described in Church (1988), DeRose (1988) and Merialdo (1990). In general, it is probably a good strategy to unbundle as many aspects of the word-sense discrimination problem as possible, and develop special purpose tools for each such aspect.

This paper will focus primarily on the aspects that are most likely to be solved with information retrieval-like methods. These methods were originally designed to discriminate documents on the basis of ‘‘subject area’’ and ‘‘topic’’, and therefore, we expect that these same methods are likely to do a fairly good job in distinguishing senses such as judicial sentences and syntactic sentences which are usually associated with different ‘‘subject areas’’ and ‘‘topics’’. These aspects of polysemy are somewhat less well understood and less studied than other aspects such as part of speech assignment.

6. Materials

6.1 Sentence Alignment

Following Brown *et al.* (1990), we begin by aligning the parallel texts at the sentence level. That is, we input a pair of texts and output a sequence of aligned regions as illustrated in Table 3. (The text used in this example was extracted from the UBS (Union Bank of Switzerland) corpus, which is available from the ACL/DCI.) The problem for the alignment program is deciding where to place the horizontal lines. In our experience, 90% of the English sentences match exactly one French sentence, but other possibilities, especially two sentences matching one (2-1) or one matching two (1-2), are not uncommon. Table 3 illustrates a particularly hard paragraph; only two of the sentences line up one for one.

Table 3: Output from Alignment Program

English	French
According to our survey, 1988 sales of mineral water and soft drinks were much higher than in 1987, reflecting the growing popularity of these products. Cola drink manufacturers in particular achieved above-average growth rates.	Quant aux eaux minérales et aux limonades, elles rencontrent toujours plus d'adeptes. En effet, notre sondage fait ressortir des ventes nettement supérieures à celles de 1987, pour les boissons à base de cola notamment.
The higher turnover was largely due to an increase in the sales volume.	La progression des chiffres d'affaires résulte en grande partie de l'accroissement du volume des ventes.
Employment and investment levels also climbed.	L'emploi et les investissements ont également augmenté.
Following a two-year transitional period, the new Foodstuffs Ordinance for Mineral Water came into effect on April 1, 1988. Specifically, it contains more stringent requirements regarding quality consistency and purity guarantees.	La nouvelle ordonnance fédérale sur les denrées alimentaires concernant entre autres les eaux minérales, entrée en vigueur le 1er avril 1988 après une période transitoire de deux ans, exige surtout une plus grande constance dans la qualité et une garantie de la pureté.

There has been quite a bit of recent work on sentence alignment, e.g., (Kay and Röscheisen, 1988), (Catizone, Russell, and Warwick, to appear), (Brown, Lai and Mercer, 1991) and (Gale and Church, 1991a). All four methods divide the two texts into sentences and then use dynamic programming to join sentences into regions as necessary subject to a local distance measure. (Kay and Röscheisen, 1988) and (Catizone, Russell, and Warwick, to appear) use a distance measure based on ''corresponding'' words such as *house* and *maison*. Unfortunately, the task of determining word correspondences is computationally expensive and consequently, this approach has not yet been scaled up to deal with a large text such as a few years of the Canadian Hansards.

(Brown, Lai and Mercer, 1991) and (Gale and Church, 1991a) use a weaker method that is easier to scale up; the distance measure ignores the words and just counts the length of the two regions. It turns out that the length of a text is highly correlated with the length of its translation. That is, a longer sentence will be translated into a longer sentence (or perhaps two sentences whose total length is not very different from the original sentence), whereas a shorter sentence will be translated into a shorter sentence. The two approaches differ in how they count the length of a region. (Brown, Lai and Mercer, 1991) count words whereas (Gale and Church, 1991a) count characters. In our experience, characters seem to work slightly better because longer words tend to be translated into longer words (or several shorter words), whereas shorter words tend to be translated into shorter words.

It is remarkable that such a simple approach works as well as it does. The method correctly aligned all but 4% of the regions. Moreover, by selecting the best scoring 80% of the corpus, the error rate dropped to 0.7%. See (Gale and Church, 1991a) for more details on the method and its evaluation.

6.2 Word Correspondences

The previous section described the first step of aligning sentences. The second step in the preparation of the testing and training materials is to find word correspondences. Gale and Church (1991b) describe a program that is intended to identify which words in the English text correspond to which words in the French text. For example, given the input sentences:

English: We took the initiative in assessing and amending current legislation and policies to ensure that they reflect a broad interpretation of the charter.

French: Nous avons pris l'initiative d'évaluer et de modifier des lois et des politiques en vigueur afin qu'elles correspondent à une interprétation généreuse de la charte.

The program, described in (Gale and Church, 1991b), would produce the following alignment:

We	took	the	initiative	in	assessing	and	amending	current
	pris		*initiative*		*evaluer*		*modifier*	

legislation	and	policies	to	ensure	that	they	reflect
lois		*politiques*		*afin*			*correspondent*

a	broad	interpretation	of	the	charter
	genereuse	*interpretation*			*charte*

In this example, 11 out of the 23 (48%) English words were matched with a French word, and 12 of the English words were left unmatched. This program has been run on the 18 million word sample of Canadian Hansards mentioned above.

We wish to distinguish the terms *alignment* and *correspondence*. The term *alignment* will be used when order constraints must be preserved and the term *correspondence* will be used when order constraints need not be preserved and crossing dependencies are permitted. We refer to the matching problem at the word level as a correspondence problem because it is important to model crossing dependencies (e.g., *sales volume* and *volume des ventes*). In contrast, we refer to the matching problem at the sentence level as an alignment problem because we believe that it is not necessary to model crossing dependencies at the sentence level as they are quite rare and can be ignored (at least for now).

6.2.1 Contingency Tables Suppose that we suspected that *house* and *chambre* were translations of one another, and we wanted to confirm this hypothesis. It turns out that *house* and *chambre* are, in fact, highly associated in this rather unusual genre. (In most other genres, *maison* would almost certainly be the preferred translation.)

Table 4 shows a two-by-two contingency table for the two words, *house* and *chambre*. Cell *a* (upper-left) counts the number of sentences (aligned regions) that contain both *house* and *chambre*. Cell *b* (upper-right) counts the number of regions that contain *house* but not *chambre*. Cells *c* and *d* fill out the pattern in the obvious way.

Table 4: A Contingency Table

	chambre	
house	31,950	12,004
	4,793	848,330

We can now measure the association between *house* and *chambre* by making use of any one of a number of association measures such as mutual information. After introducing a number of refinements, we were able to scale the process up to handle a large fraction of the Hansards. The program found correspondences for about 85% of the content words.[8] The program tended to error on the safe side; 96% of the correspondences

8. In fact, only about 90% of the content words do have correspondences in the other language. The remaining 10% are counter-examples to the word-for-word approximation.

produced by the program were correct.

In fact, many of the results reported here used a somewhat simpler method of finding word correspondences. Example sets were selected by specifying the English noun and a set of possible French translations. A program picked out all of the cases in which the given noun and a translation occurred in about the same relative positions within one aligned region.

7. Word-Sense Disambiguation Problems

Consider, for example, the word *duty* which has at least two quite distinct senses: (1) a tax and (2) an obligation. Three examples of each sense are given below in Table 5.

Table 5: Sample Concordances of *duty* (split into two senses)

Sense	Examples (from Canadian Hansards)
tax	fewer cases of companies paying >duty< and then claiming a refund
	and impose a countervailing >duty< of 29,1 per cent on candian exports of
	the united states imposed a >duty< on canadian saltfish last year
obligation	it is my honour and >duty< to present a petition duly approved
	working well beyond the call of >duty< ? SENT i know what time they start
	in addition , it is my >duty< to present the government 's comments

The classic word-sense disambiguation problem is to construct a means for discriminating between two or more sets of examples such as those shown in Table 5.

After we have discussed this problem, we will turn to a number of additional problems. It would be useful to be able to construct a means of recognizing additional examples of one sense given one set of examples of that sense. This would be useful when one had a few examples of one sense and desired more to achieve a greater accuracy. Locating examples of a particular sense by hand is slow, so we are interested in finding methods that might reduce the effort required to tag a large corpus with senses.

It would also be useful to be able to decide whether two sets of examples should be lumped together into a single sense or whether they should be split into two senses. A slightly harder task is to start with a single set of examples and decide wether it should be split into two senses. The ultimate task is: given a single set of examples, partition the set into subsets, where each subset corresponds to a distinct sense.

8. Discrimination Problems

We regard the word-sense disambiguation problem as a discrimination problem, not very different from problems such as author identification and information retrieval. In author identification and information retrieval, it is customary to split the process up into a testing phase and a training phase. During the training phase, we are given two (or more) sets of documents and are asked to construct a discriminator which can distinguish between the two (or more) classes of documents. These discriminators are then applied to new documents during the testing phase. In the author identification task, for example, the training set consists of several documents written by each of the two (or more) authors. The resulting discriminator is then tested on documents whose authorship is disputed. In the information retrieval application, the training set consists of a set of one or more relevant documents and a set of zero or more irrelevant documents. The resulting discriminator is then applied to all documents in the library in order to separate the more relevant ones from the less relevant ones. In the sense disambiguation case, the 100-word context surrounding instances of a polysemous word (e.g., *duty*) are treated very much like a document.

There is an embarrassing wealth of information in the collection of documents that could be used as the basis for discrimination. To date, most researchers have tended to treat documents as ‘‘merely’’ a bag of words, and have generally tended to ignore much of the linguistic structure, especially dependencies on word order and correlations between pairs of words. The collection of documents can then be represented as a term by document matrix, where each cell counts the number of times that a particular term appears in a particular document. Since there are $V \approx 100,000$ terms, the term by document matrix contains a huge amount of information, even allowing for the fact that the matrix is quite sparse and many of the cells are empty.

It is natural to take a Bayesian approach to these discrimination problems. Mosteller and Wallace (1964, section 3.1) used the following formula to combine new evidence (e.g., the term by document matrix) with prior evidence (e.g., the historical record) in their classic authorship study of the Federalist Papers.

$$final\ odds = (initial\ odds) \times (likelihood\ ratio)$$

For two groups of documents, the equation becomes

$$\frac{P(class_1)}{P(class_2)} = \frac{p(class_1)}{p(class_2)} \times \frac{L(class_1)}{L(class_2)}$$

where P represents a final probability, p represents an initial probability, and L represents a likelihood. Similar equations can be found in textbooks on information retrieval (e.g., Salton (1989), equation 10.17).

The initial odds depend on the problem. In the author identification problem, for example, the initial odds are used to model what we know about the documents from the various conflicting historical records. In the information retrieval application, the user may have a guess about the fraction of the library that he or she would expect to be relevant; such a guess could be used as the prior. It is often the case that the prior probability will not have very much influence on the outcome, which is fortunate, since the prior can sometimes be difficult to estimate.

It is common practice to decompose the likelihoods into a product of likelihoods over tokens in the document (under appropriate independence assumptions):

$$\frac{L(class_1)}{L(class_2)} \approx \prod_{tok\ in\ doc} \frac{Pr(tok|class_1)}{Pr(tok|class_2)}$$

The crucial ingredients for this calculation are the probabilities of each term in the vocabulary *conditional* on the document being from a given class. These conditional probabilities have been computed in a number of different ways depending on the application and the study.

For two senses, the Bayesian equation mentioned above becomes:

$$\frac{P(sense_1)}{P(sense_2)} = \frac{p(sense_1)}{p(sense_2)} \times \frac{L(sense_1)}{L(sense_2)}$$

where p, P and L are the initial probability, the final probability and likelihood, as before. The initial probabilities are determined from the overall probabilities of the two senses in the corpus. As other large dimension discrimination problems, the likelihoods are decomposed into a product over tokens:

$$\frac{L(sense_1)}{L(sense_2)} \approx \prod_{tok\ in\ context} \frac{Pr(tok|sense_1)}{Pr(tok|sense_2)}$$

As mentioned above, this model ignores a number of important linguistic factors such as word order and collocations (correlations among words in the context). Nevertheless, there are $2V \approx 200,000$ parameters in the model. It is a non-trivial task to estimate such a large number of parameters, especially given the sparseness of the training data. The training material typically consists of approximately 12,000 words of text (100 words words of context for 60 instances of each of two senses). Thus, there are more than 15 parameters to be estimated from for each data point. Clearly, we need to be fairly careful given that we have so many parameters and so little evidence.

The conditional probabilities, $Pr(tok|sense)$, can be estimated in principle by selecting those parts of the entire corpus which satisfy the required conditions (e.g., 100-word contexts surrounding instances of one sense of *duty*), counting the frequency of each word, and dividing the counts by the total number of words satisfying the conditions. However, this estimate, which is known as the maximum likelihood estimate (MLE), has a number of well-known problems. In particular, it will assign zero probability to words that do not happen to appear in the sample. Zero is not only a biased estimate of their true probability, but it is also unusable for the sense disambiguation task (and for quite a number of other applications). In addition, MLE also produces poor estimates for words that appear only once or twice in the sample. In another application (spelling correction), we have found that poor estimates of context are worse than none; that is, at least in this application, we found that it would be better to ignore the context than to model it badly with something like MLE (Gale and Church, 1990).

The proposed method uses the information from the entire corpus in addition to information from the conditional sample in order to avoid these problems. We will estimate $Pr(tok|sense)$ by interpolating between word probabilities computed within the 100-word context and word probabilities computed over the entire corpus. For a word that appears fairly often within the 100-word context, we will tend to believe the local estimate and will not weight the global context very much in the interpolation. Conversely, for a word that does not appear very often in the local context, we will be much less confident in the local estimate and will tend to weight the global estimate somewhat more heavily. The key observation behind the method is this: the entire corpus provides a set of well measured probabilities which are of unknown relevance to the desired conditional probabilities, while the conditional set provides poor estimates of probabilities that are certainly relevant. Using probabilities from the entire corpus thus introduces bias, while using those from the conditional set introduce random errors. We seek to determine the relevance of the larger corpus to the conditional sample in order to make this trade off between bias and random error.

The interpolation procedure makes use of a prior expectation of how much we expect the local probabilities to differ from the global probabilities. Mosteller and Wallace ‘‘expect[ed] both authors to have nearly identical rates for almost any word’’ (p. 61). In fact, just as they had anticipated, we have found that only 2% of the vocabulary in the Federalist corpus has significantly different probabilities depending on the author. Moreover, the most important words for the purposes of author identification appear to be high frequency function words. Our calculations show that *upon, of* and *to* are strong indicators for Hamilton and that *the, and, government* and *on* are strong indicators for Madison. These are all high frequency function words (at least in these texts), with the exception of *government,* which is, nontheless, extremely common and nearly devoid of content.

In contrast, we expect fairly large differences in the sense disambiguation application. For example, we find that the tax sense of *duty* tends to appear near one set of content words (e.g., *trade* and *lumber*) and that the obligation sense of *duty* tends to appear near quite a different set of content words (e.g., *honour* and *order*), at least in the Hansard corpus. Approximately 20% of the vocabulary in the Hansards has significantly different probabilities near *duty* than otherwise. In short, the prior expectation depends very much on the application. In any particular application, we set the prior by estimating the fraction of the vocabulary whose conditioned probabilities differ significantly from the global probabilities.

9. The Interpolation Procedure

Let the entire corpus be divided into a conditional sample of size n and the residual corpus (the entire corpus less the conditional sample) of size $N >> n$. Let a be the frequency of a given word in the conditional sample, and A its frequency in the residual corpus. Either of these frequencies may be zero, but not both. Let π represent the conditional probability of the word. Before knowing the frequency of the word in either sample, we could express our ignorance of the value of π by the *uninformative distribution*:

$$B^{-1}(½,½)\,\pi^{-½}(1-\pi)^{-½}$$

where $B(x,y)$ is the Beta function. Several variations of the method can be based on variations in the uninformative distribution. If A additional observations out of N, relevant to the determination of π, were made the distribution expressing our knowledge would become

$$B^{-1}(A+½,N-A+½)\,\pi^{A-½}(1-\pi)^{N-A-½}$$

While we have A out of N observations of the word in question in the residual corpus, we do not know their relevance. Thus we set as our knowledge before observing the conditional sample the distribution:

$$p(\pi) \;=\; rB^{-1}(A+½,N-A+½)\,\pi^{A-½}(1-\pi)^{N-A-½} \;+\; (1-r)\,B^{-1}(½,½)\,\pi^{-½}(1-\pi)^{-½}$$

where $0 \le r \le 1$ is the relevance of the residual corpus to the conditional sample. When $r=0$, this gives the uninformative distribution, while if $r=1$, it gives the distribution after observing the residual corpus. Another way of reading this is that with probability r we are expecting an observation in line with the residual corpus, but that with probability $1-r$ we won't be surprised by any value.

The joint probability of observing a out of n instances of the word in question in the conditional sample and that the conditional probability is π is then

$$p(\pi,a) \;=\; \begin{bmatrix} n \\ a \end{bmatrix} \left\{ rB^{-1}(A+½,N-A+½)\,\pi^{A+a-½}(1-\pi)^{N-A+n-a-½} \;+\; (1-r)\,B^{-1}(½,½)\,\pi^{a-½}(1-\pi)^{n-a-½} \right\}$$

We then form

$$p(a) \;=\; \int_0^1 p(\pi,a)\,d\pi$$

and

$$p(\pi|a) \;=\; p(\pi,a)/p(a)$$

which is then integrated to give

$$E(\pi|a) \;=\; \int_0^1 \pi p(\pi|a)\,d\pi$$

$$= \frac{r\dfrac{B(A+a+1½,N-A+n-a+½)}{B(A+½,N-A+½)} \;+\; (1-r)\,\dfrac{B(a+1½,n-a+½)}{B(½,½)}}{r\dfrac{B(A+a+½,N-A+n-a+½)}{B(A+½,N-A+½)} \;+\; (1-r)\,\dfrac{B(a+½,n-a+½)}{B(½,½)}}$$

This can be approximated in various ways, but it is practical to calculate it directly using the relationship

$$B(x,y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

The parameter *r*, which denotes the relevance of the residual corpus to the conditional sample, can be estimated in various ways. Its basic interpretation is the fraction of words that have conditional probabilities close to their global probabilities (as estimated from the residual sample). Thus given a set of estimates of conditional probabilities, one can estimate *r* as the fraction of them which lie within a few standard deviations of the corresponding global probabilities. This estimate is performed using the words which are observed in the conditional sample. Alternatively *r* can be regarded as a free parameter of the method and adjusted to produce optimal results on a specific task. Although it could be varied for each word, we have used $r = 0.8$ for all words in the sense disambiguation application, and $r = 0.98$ for all words in the author identification application.

10. Example of the Interpolation Procedure

Table 6 gives a sense of what the interpolation procedure does for some of the words that play an important role in disambiguating between the two senses of *duty* in the Canadian Hansards. Recall that the interpolation procedure combines local probability estimates with global probability estimates. The local estimates are obtained from the conditioned sample and are therefore considered more relavent; the global probability estimates are obtained from the entire corpus and are therefore less relevant, but also less subject to sparse data issues.

The conditioned samples are obtained by extracting a 100-word window surrounding each of the 60 training examples. The training sets were selected by randomly sampling instances of *duty* in the Hansards until 60 instances were found that were translated as *droit* and 60 instances were found that were translated as *devoir*. The first set of 60 are used to construct the model for the tax sense of *duty* and the second set of 60 are used to construct the model for the obligation sense of *duty*.

The column labeled ‘‘freq’’ shows the number of times that each word appeared in the conditioned sample. For example, the count of 50 for the word *countervailing* indicates that *countervailing* appeared 50 times within the 100-word window of an instance of *duty* that was translated as *droit*. This is a remarkable fact, given that *countervailing* is a fairly unusual word. It is much less surprising to find a common word like *to* appearing quite often (228 times) in the other conditioned sample.

The second column (labeled ‘‘weight’’) models the fact that 50 instances of *countervailing* are more surprising than 228 instances of *to*. The weights for a word are its log likelihood in the conditioned sample compared with its likelihood in the global corpus. The first column, the product of these log likelihoods and the frequencies, is a measure of the importance, in the training set, of the word for determining which sense the training examples belong to. Note that words with large scores do seem to intuitively distinguish the two senses, at least in the Canadian Hansards.

There are obviously some biases introduced by the unusual nature of this corpus, which is hardly a balanced sample of general language. For example, the set of words listed in Table 6 under the obligation sense of *duty* is heavily influenced by the fact that the Hansards contain a fair amount of boilerplate of the form: ‘‘Mr. speaker, pursuant to standing order..., I have the honour and duty to present petitions duly signed by... of my electors...’’.

Table 6 gives the 15 words with the largest product (shown as the first column) of the model score (the second column) and the frequency in the 6000 word training corpus (the third column).

Table 6: Selected Portions of Two Models

tax sense of *duty*				obligation sense of *duty*			
weight*freq	weight	freq	word	weight*freq	weight	freq	word
285	5.7	50	countervailing	64	3.2	20	petitions
111.8	4.3	26	duties	59.28	0.26	228	to
99.9	2.7	37	u.s	56.28	0.42	134	|
73.1	1.7	43	trade	51	3	17	petition
70.2	1.8	39	states	47.6	2.8	17	pursuant
69.3	3.3	21	duty	46.28	0.52	89	mr
68.4	3.6	19	softwood	37.8	2.7	14	honour
68.4	1.9	36	united	37.8	1.4	27	order
58.8	8.4	7	rescinds	36	2	18	present
54	3	18	lumber	33.6	2.8	12	proceedings
50.4	4.2	12	shingles	31.5	3.5	9	prescription
50.4	4.2	12	shakes	31.32	0.87	36	house
46.8	3.6	13	35	29.7	3.3	9	reject
46.2	2.1	22	against	29.4	4.2	7	boundaries
41.8	1.1	38	canadian	28.7	4.1	7	electoral

By interpolating between the local and global probabilities in this way, we are able to estimate considerably more parameters than there are data points (words) in the training corpus. The interpolation procedure assumes that one selection of natural language is roughly similar to another. In this way, it becomes feasible to estimate the $2V \approx 200{,}000$ parameters, one for each word in the vocabulary and each of the two senses. We are, of course, assuming that words are pairwise independent and that all of the correlations between words are zero.

10.1 Context Width

How wide a context is relevant to distinguishing senses? Previous work has all focused on words that are quite nearby, with Black's use of an entire concordance line being the largest context used. This approach appears to stem from Kaplan's (1950) observation that ''a context consisting of one or two words has an effectiveness not markedly different from that of the whole sentence,'' when people do the disambiguation task. The same observation, that people can do the disambiguation task effectively with just one or two words of context, is demonstrated in a more recent paper by Choueka and Lusignam (1985). However, as has been found in chess playing programs, attempting to model the way people do things may not be the best way for a computer to do the same task.

Since we are concerned with disambiguation in a large corpus, we can have virtually any amount of context that we want. The question is its relevance and utility. The following figure shows that in the Hansards, context is *relevant* to disambiguating nouns, up to ten thousand words away.

Remote Context is Informative

Figure 1. The horizontal axis shows the distance of context words from an ambiguous word, while the vertical scale shows the percent correct when using ten context words at the specified distance in doing the disambiguation. The vertical lines show the mean and standard deviation of mean for six disambiguations. With two equiprobable choices, 50 percent represents chance performance. Performance remains above chance for ten word contexts up to ten thousand words away from the ambiguous word.

To make Figure 1 we trained models for each of the six chosen words on sixty examples of each of the two main senses, using a context window of fifty words to either side of the ambiguous word. For the remaining test sample of 90 words for each sense, we scored the models on the words occurring in the intervals [–w–5,–w] and [w, w+5] words away from the ambiguous word. This test thus asks, if you did not know any of the intervening words, would the ten words at least *w* words away be useful for disambiguation? The answer is ‘‘yes’’ for any *w* less than ten thousand, for the Hansards. We found this surprising, given the focus of previous quantitative models on nearby context, although it can be seen to relate to some kind of maximum discourse length within the floor debates. The result could be different for verbs or adjectives.

Of course, the information at a remote distance may just duplicate information from the nearby context. The question for building a practical model is what the *marginal* value of information at some distance is. Figure 2 addresses this question.

Wide Contexts are Useful

Figure 2. The horizontal axis shows the maximum distance of context words from an ambiguous word, while the vertical scale shows the percent correct when using all context words out to the specified distance in disambiguation. While performance rises very rapidly with the first few words, it clearly continues to improve through about twenty words, and is not worse by fifty words.

To make Figure 2, we again trained models for each of the six chosen words on sixty examples of each of the two main senses, using a ±50 context window. For the remaining test sample of 90 words for each sense, we scored the models on the words occurring in the intervals [–w,–1] and [1,w] words away from the ambiguous word. This test thus asks, given that you do know all the words out to some distance, what is the value of a few more words further out. The largest contribution is, not surprisingly, from the closest words. However, the context is clearly marginally valuable out to twenty words, and possibly valuable as far as fifty words.

We have use fifty words of context as the standard for our models. The widest previous context used was about ±six words. The improvement in performance from 85% to ??? to 90%??? for disambiguating two equiprobable senses represents and important gain from using fifty word context over six word context.

10.2 Number of Training Examples

Previous work has not controlled the number of examples of each sense used for training, but has accepted the naturally occurring frequencies of each sense. We expect the quality of the models built to depend on the number of examples used for training, and we would like to control this source of variability as we study other factors. We would also like to know what performance one can expect from models trained on relatively few examples, since most types and most senses do not have many examples.

Figure 3 shows the effect of varying the number of training examples.

Just a Few Training Examples do Surprisingly Well

Figure 3. The horizontal axis shows the number of examples used in training while the vertical scale shows the mean percent correct in six disambiguations. The performance increases rapidly for the first few examples, and seems to have reached a maximum by 50 or 60 examples.

As few as three examples will give models that achieve 75 percent accuracy while ten give 80 percent accuracy. Thus, while accuracy would suffer some, models of some utility could be built for senses occurring just a few times in a corpus.

10.3 Training with Errors

Previous authors have not considered the effect of errors in the training materials. Yet if training materials are to be collected on a large scale, there will need to be considerable automatic assistance, and there will clearly be some errors. Also, if models can be built from materials with errors, then an iterated selection of materials by the model may allow collecting a better training set.

We constructed training sets with errors by deliberately making 10, 20, or 30 percent of the training materials for a given sense be examples from the alternative sense. Fifty examples were used for training, with a ±fifty word context. Table 7 shows the mean percent error for our six nouns.

Table 7: Errors in the Training Materials are Acceptable

% errors	0	10	20	30
% coverage				
50	1.4	4	8	16
80	7	8	12	19
100	10	12	17	22

Table 7 shows three levels of contamination and three levels of coverage. Fifty percent coverage means selecting the half of the test examples for which the discrimination score is the largest. Two observations on this table are important. First, at 10 percent errors input, the output errors have only increased from 10 percent to 12 percent. Thus we can accommodate up to about ten percent errors with little degradation of performance. Second, at fifty percent coverage, input errors of twenty to thirty percent result in about half as many errors on output. Therefore if one had obtained a set of examples with no more than twenty to thirty percent, one could iterate example selection just once or twice and have example sets that had less than ten percent errors.

10.4 Factors in Evaluating Performance

The reports of performance in previous work each use different conventions and conditions so that comparison is not easy. Kelly and Stone (1975) report ‘‘kappa,’’ and do not give the equation for this measure despite having a book length treatment. They describe kappa as ‘‘simply a refinement of raw percent agreement, to correct for chance agreement.’’ A reference is given, but we have not obtained a copy of it. Their Figure 6 of Chapter III shows kappa for ‘‘semantic’’ disambiguations, the sort of disambiguation we have studied. The performance reported is given in the table as 49 cases with kappa between 0 and 0.7, 25 cases with kappa between 0.7 and 0.9, and 18 cases with kappa between 0.9 and 1.0. For a rough interpretation of this data, we took an average over the groups, representing each group by its midpoint. The mean kappa is thus 0.59. They report that the raw percent correct overall is ‘‘just under 90 percent,’’ while the same table shows that the overall kappa is .69. A plausible guess at the percent correct for the semantic distinctions is .59*.9/.69=.77. There is no indication of how many senses were disambiguated for each word, although two is the most likely. It is clear that all of the words in the test data were attempted, so coverage was 100 percent.

Black (1988) reports several measures, including raw percent correct, for each word. The best of the three methods that he studied achieved 75 percent correct averaged over the five words. Four of the words had four senses while one had three senses, so this was a deliberately difficult set of words. Coverage was again 100 percent.

Dagan (1991) reports on two small experiments, one on 105 ambiguous Hebrew words and one one 54 ambiguous German words. Combining the figures he gives for the two, his methods made a decision for 105 of the 159 cases, for a 66 percent coverage. They were correct in 91 of the 105 cases, for an 87 percent accuracy at the 66 percent coverage. It is not clear how many senses per word were considered; two is most likely.

The following figure shows the accuracy versus coverage (similar to the precision versus recall figures of information retrieval) for our methods on two equiprobable senses.

Errors Fall for Decreased Coverage

Figure 4. The horizontal axis shows coverage while the vertical scale shows error rate. The error rate drops quickly for small decreases in coverage, because most of the errors occur for cases with little information.

This figure shows that it is important to state a coverage in comparing accuracies, because the error rate drops quickly for small decreases in coverage. At 66 percent coverage, the error rate is about a quarter of its value at 100 percent coverage.

Previous work states error rates for naturally occurring frequencies of senses rather than for controlled equiprobable frequencies as we do. This conflates the value of the information from the method and the information one starts with by knowing the frequencies of the senses. However, when the methods are applied, both sources of information would certainly be used, thus some estimate of the power of the combined sources is appropriate. The following figure shows the error rate for 100 percent coverage as a function of the probability of the major sense.

Overall Errors Depend on Prior Information

Figure 5. The horizontal axis shows the prior fraction for the major sense while the vertical scale shows the error rate for the combination of prior and context. The dotted lines show minimum of the two errors; the combination should do better than either. Unless the information from the prior is comparable to that from the context, there is little improvement over the context alone.

This figure is constructed by using the data from our six words as hypothetically coming from words whose major sense has the probability P_M while the minor sense has the probability p_m. The test examples are scored by adding $\log(P_M/p_m)$ to the previously discussed likelihoods. After counting the number of correct test cases for each sense (the fraction correct C_M for the ''major'' sense will be higher than the fraction c_m for the ''minor'' sense), they are averaged by calculating $P_M C_M + p_m c_m$. For values of P_M other than 0.5, two such weighted averages are averaged, one from taking each sense as the ''major'' sense.

The figure shows the minimum of the errors from the prior and the context as a dotted line. The combination of the evidence should do better than either alone, as indeed it does. The figure shows, however, that unless the information from the prior is about as good as that from the context, there will be little gained by the combination. The degree of dominance of the major sense in a set of ambiguous words will depend on how the words are chosen. If the list of words is generated outside the corpus, many will be nearly unambiguous in the corpus, so the dominance will be quite high. Our six words were chosen from ords which are definitely ambiguous in the Hansards, and which have a well populated minor sense. Nevertheless, the average dominance for the six is .77; we expect most sets of words to have higher dominance. At a dominance of .77, the methods give 92% accuracy at 100% coverage when context and prior information are combined.

The previous figure was generated by a non-literal use of the data, that is, we could study the combination of prior and context information for other than the actual values of the priors for the words we have considered. Another non-literal use of our data is to make artificial ''words'' with known different senses, or example sets known not to be different. We can make artificial ''words'' by combining arbitrary sets of senses, such as the drug/drogue set with the land/pays set. We can make sets of examples with no known difference by making a random split of one of our larger sense sets, such as the land/terre set. In making combinations, we have used senses from additional English words. In research that is limited by the availability of training materials, these combinations are a new way to extend given materials. We believe it is a valid extension, because the differences that we are studying are large enough that they could easily have different words. In fact, in French they do have different words.

Although we did not find words with more than 150 examples of a third or fourth sense translated by a different French words we have studied some synthetic words composed by joining pairs of the words we have studied. These synthetic words thus all have four senses. The methods give an average accuracy of 82 percent correct when applied to (drug+duty, duty+land, land+sentence, sentence+position, position+language, and language+drug). Not that although the percent correct has dropped, the information provided by the method has risen. A simple approximation to the information in an n-way choice in which f_C is the fraction correct is:

$$log(n) \ + \ f_C \log(f_C) \ + \ (n-1) \ \frac{f_C}{(n-1)} log(1-f_C)$$

where the logs are to the base 2. The equation is exact for $n=2$, and assumes an equal distribution for wrong answers for $n>2$. The information from 90 percent correct on a two way choice is .53 bits, while the information from 82 percent correct on a four way choice is about 1.32 bits.

While we have tried in this section to compare performance with previous reports, the more important comment is that coverage and number of senses disambiguated affect performance strongly. It should also be apparent that what kinds of sense differences are chosen for study will affect performance, and that there is no way to compare these differences yet.

11. Locating Additional Examples of a Sense

Given a few examples of a sense of a word, we want to find some additional senses.

An aid for this task can be built as a variation of the Bayesian discrimination model. The discrimination model calls for comparing two sets of conditional probabilities via:

$$\frac{P(sense_1)}{P(sense_2)} \ = \ \frac{p(sense_1)}{p(sense_2)} \times \frac{L(sense_1)}{L(sense_2)}$$

$$= \ \frac{p(sense_1)}{p(sense_2)} \frac{\prod_i p(word_i \ | \ sense_1)}{\prod_i p(word_i \ | \ sense_2)}$$

Since this task deals with only one sense, and just one set of conditional probabilities, rather than two, we need to modify the approach. We can compare the likelihood that a word comes from the neighborhood of the exemplified sense to the likelihood that it comes from some random place in the corpus. This is accomplished by using the global probabilities $p(word_i)$ in the above equation instead of $p(word_i \ | \ sense2)$. When the resulting score is positive, then the neighborhood of the word in question is more like the neighborhoods of the exemplified sense than it is like the global distribution.

The more training examples there are, the better we would expect to do at picking out other examples of the desired sense, but the fewer examples we need, the more useful the procedure. The following figure shows accuracy for the entire set of examples with positive score and for the ten percent with the best scores.

Additional Examples can be Found Easily

Figure 6. The horizontal axis shows the number of examples used for training, while the vertical scale shows precision or recall when one sense alone is compared to the global probabilities. The solid line shows precision (examples of desired sense retrieved/all examples retrieved), while the dotted line shows recall (examples of desired sense retrieved/all examples of desired sense). Precision is high however many examples are used for training, while recall rises as the representativeness of the training examples increases.

The figure is constructed using examples trained on ±50 word contexts, with the same context used for recognition. This task is analogous to an information retrieval task, to find examples of desired documents (the desired sense) out of a given library (the set of all the words in question). Therefore we have shown the standard IR measures of precision and recall in the figure. It is remarkable that precision is high even when only a few examples of the desired sense are presented. This makes the technique of definite use for finding additional examples of a given sense. However, the low recall for this condition warns that the retrieved examples will probably be over specific, closely resembling the training cases and not representing the sense as a whole.

12. Testing Whether Two Sets of Examples Differ in Sense

Given two sets of examples, we want to make a decision to leave them separate or to join them.

This task occurs in several settings. One setting arises from the multiplicity of French words used to translate most English words. For instance, Table 8 shows three examples of *sentence* that were translated with *peine* and three examples of *sentence* that were translated with *sentence*.

Table 8: Concordances for ‘‘Sentence’’

sentence / peine

it seems pretty clear that he did not serve the >sentence< that was imposed upon him
in other words , we are increasing his >sentence< . SENT he was given a sentence of 10 years
after one-third of a >sentence< has been served . SENT parole is granted because

sentence / sentence

48 are serving a life >sentence< for first degree murder
he can consider mercy . SENT if the >sentence< is for capital punishment ,
when somebody is given a three-year >sentence< , indeed he should not be released after serving six months .

It seems obvious that these two groups should be joined, but can we test this automatically?

As another example, Table 9 gives a number of examples of *duty*, separated by the previous word, either *countervailing duty* or *my duty*.

Table 9: Concordances for ‘‘Duty’’

countervailing duty

the antidumping and countervailing >duty< statutes of the parties
there must be new rules applied to countervailing >duties< and subsidies
the government will open the door to a countervailing >duty<

my duty

it is not my >duty< to defend them , but i think that to be perfectly fair
mr . speaker , it is my >duty< and honour to present to the house

In this case, it is clear that the groups should not be joined, but can we determine this automatically?

A number of tests suggest themselves. We could build models of each set and see how well the models differentiate the sets, or we could build a model of either one set and see how highly the other set scores. We could measure the effect of the models by means or medians of scores or by the percent of cases that were correctly classified. We gathered data on these nine procedures, but quickly rejected counting correctly classified cases, which was a coarser measure than mean or median scores. We also quickly dropped tests based on building two models as they did not perform as well as those based on one model. The mean and median scores performed just about as well, and we focused on the mean scores. Thus we discuss two tests here, based on modeling the larger group or the smaller group and scoring the other.

The procedure is thus similar that of the previous section: build a model of one group of examples compared to the global population, score each example in the other group using the model and compare the mean score to a standard. The standard is derived by applying the procedure to two known conditions, the groups being known to be different, and known to be the same. Groups known to be the same were constructed by sampling from one English/French set, and maximally different groups were constructed by taking one E/F set from each of two different Engish words. The larger group was held at 64 examples while the smaller group was varied from 8 to 64 examples by factors of 2. One hundred trials were made for each of the same sense decisions and for each of the different sense decisions, resampling the original groups for each trial. Building a model on the smaller group and scoring it on the larger group turned out to be more accurate. The following table shows its performance:

Table 9: Different Contexts Can Be Identified Automatically

examples of minor sense	errors on same	errors on different
8	18	2
16	18	1
32	18	1
64	18	1

The mean scores for same groups are of course higher than scores for the different groups; we took as the decision point the mean of these two means. A higher mean on one particular test was scored as the same sense while a lower mean score than this decision point was scored as different senses. The table shows that groups of examples from different senses can be distinguished quite reliably. However, when the groups are actually the same, they appear to be different about one time in five. Thus judgements that two groups are the same will have very few instances of different groups, but the set of judgements that call for a difference will need to be examined to find the cases that are actually the same. Experience using this technique on groups differing in French tag, and in groups differing in some collocate bears out this conclusion: when the groups are marked as the same, then they are the same. But if they are marked different, they may still be the same.

13. Testing Whether One Set of Examples Contains Subsets Differing in Sense

A harder task than the previous is to determine whether one set of examples needs to be split into two or more sets. The previous section describes how to test whether two sets of examples should be joined. Thus one way to test whether a single set needs to be split is to split it by some criterion other than the remote context, and ask whether there are distinct senses represented among the fragments.

One way of splitting a set of examples that we have studied is to use the few words nearest to the target word to form subgroups. We can then ask whether they can be distinguished by more distant context.

The procedure we use is as follows. Use a context of ±two words to build a Bayesian model. Let w_i be the model weight for the i^{th} word in the vocabulary, $i = 1, \cdots V$. Let f_i be the raw frequency of the i^{th} word. Then sort the vocabulary by $f_i W_i$. This is the cumulative weight for each word in the vocabulary in recognizing the training sample. Large products indicate important words. While this product has several applications, here we just want to select the top $k \approx 10$ words that are important in a narrow window model. These are important collocates of the target word.

Our first observation is that these collocate split groups are almost always entirely one sense. For instance, the first ten collocates for drug are prescription, prices, illicit, food, abuse, generic, price, paraphernalia, alcohol, and trafficking. The sets formed by taking these collocates are all one sense or the other of drug, and have no mixture. In the sixty sets formed by taking the top ten collocates for each of our six nouns, we found just two or three cases that were not pure, and they were over 90 percent in purity.

Our second observation is that each sense is exemplified by some collocate group within the first ten groups for each of our six nouns. In the drug example, for instance, prescription clearly identifies medicinal drugs and illicit identifies, well, illicit drugs. Food is in the list because in the phrase ‘‘Food and Drug Act’’ drug is translated by drogue.

Unfortunately, the rejoining techniques always leave several groups, even when just one sense is split. However, the technique is still quit useful in automating this task, because a person can tell just by looking at the collocates what senses are involved, and can easily group the collocates to form initial training sets.

14. Constructing Training Sets

One method of doing sense discrimination is to build a model for each word, or for each sense of each word, training the models on sets of examples of the given senses. Given the ability to build such models from training sets, the method still requires sets of training sets. To date research has been done using a few words for which the training sets could be constructed by hand. The discrimination methods alone do not solve the sense discrimination problem, because the hand construction of training material does not scale up. This section sets outs several possible machine aids for the construction of training sets. None is totally automatic.

We call the first method ‘‘cold tagging.’’ It consists of using *all* the examples of a word as the training set. Given the results cited above on training from a set of examples with errors, then cold tagging will give a useful model of the major sense of a word provided that the major sense accounts for at least sixty to seventy percent of the uses. The examples selected by the initial model can be used to iterate, training another model on a set with less spurious examples. The human judgement required is whether the word has such a predominant use.

A second cold tagging from the examples rejected by the model for the primary sense does not usually produce a model for one sense. However, some of the other methods described below can be used on this residue.

Another method was discussed in the previous section. It is possible to split the set of all examples on the collocates of the target word and then to rejoin the collocates. The methods here are reliable when they call for joining two such groups, but they leave too many groups separate. The human judgement required is thus to join groups of examples that have disjoint collocates. The method typically creates classes containing about half of the examples. After building models on the classes formed, the remainder of the examples can be scored. Additional methods are needed for the examples which do not fall into any of the classes so formed.

Models built on one corpus can be used to select training sets for models on another corpus. We applied the models trained on Hansard examples to the AP wire corpus. The precision of the models on the AP was far below that on the Hansard. However, by selecting the highest scoring examples for a given model, the error rate is usually low enough to use the resulting examples as a training set with errors. The following table shows error rates for the top quarter of examples selected from the AP by a model trained on the Hansards.

Table 10: Models Can Usually be Transferred between Corpora

word	quality examples		errors (percent)	
	major	minor	major	minor
drug	46	2331	2	12
duty	41	12	0	0
land	961	50	2	90
language	60	88	2	55
position	106	60	33	23
sentence	614	60	0	44

The errors are calculated for a random sample of the selected examples, selecting the maximum of 60 or the number of examples. The table shows that in each of the six cases, the model for the major sense in the Hansards can generate a set of AP training examples with thirty percent error or less. However, this was only true for three of the six models for minor senses in the Hansards. Human judgement would be required to supervise this transfer process, marking some groups for additional techniques. For instance, using the AP model for the major sense to subtract examples from the set of minor sense examples improves its quality.

Known synonyms can assist in building models for minor senses. In a small experiment, we considered the targets *tongue*, *duty*, and *land*, and the synonyms *language*, *tax*, and *country*. In the Hansard corpus, we constructed cold tag models of each of these six words. We then applied the synonym cold tag model to (a) the entire set of target examples, and (b) the set of target examples which scored negatively on their own cold tag model. The following table shows the results.

Table 11: Known Synonyms can Help Build Training Sets

synonym	target	target sense percent	synonym model accuracy all target	synonym model accuracy not major
language	tongue	50	90	75
tax	duty	50	90	75
country	land	25	50	85

The table suggests that a sense as predominant as fifty percent can have a set of training examples selected by applying a model for a synonym. However, the synonym model is not strong enough if the target sense is a minor one. In this case, however, it may be useful to run the synonym model on the residue from the cold tag model for the target word. On these few examples, this gave a training set with acceptably few errors in all three cases. Human judgement would be needed to determine whether application to the entire set, or to the residue would be a better option. If a network of synonyms were constructed from one corpus, there would be the prospect of quite an automatic training of examples on another corpus using direct models to construct training sets and models of known synonyms to validate the directly constructed set, or to flag them for human attention if they failed.

As a final fallback, the methods described here can be used to make an incremental tagger that requires only one third as many judgements from a person as compared to reviewing all examples. The procedure is simple. The computer selects an example at random to start, and the person assigns it a class. The computer than builds a model of the class from the one example. At any time, the computer selects its next example from among those that score negatively on all models, so as to ask about the examples that it cannot classify well. This preselection of examples reduces the number that the person needs to examine, by a factor of about three.

15. Conclusions

Word sense disambiguation has been recognized as a major problem in natural language processing research for over forty years. One contributory factor to the slowness of progress has been the vagueness of the notion of a sense. In fact, distinctions marked as sense difference in dictionaries are based on a multiplicity of factors, such as part of speech, number of complements, syntactic features, etymology, capitalization, and topic. Any one dictionary has some unknown mixture of these and other criteria as goals, and it therefore seems senseless to try to reproduce the exact distinctions made by any one dictionary. Our most fundamental suggestion on sense discrimination is that the problem should be analysed into component subproblems. The work described here relates to sense differences that correspond to topic differences.

The scale of sense discrimination problems has also deterred progress. While both qualitative or symbolic and quantitative methods have been tried repeatedly, each has depended on hand prepared materials. The qualitative methods have depended on hand prepared semantic networks and the quantitative methods have depended on hand tagged training materials. We have achieved considerable progress recently by taking advantage of a new source of testing and training materials. Rather than depending on small amounts of hand-labeled text, we have been making use of relatively large amounts of parallel text, text such as the Canadian Hansards, which are available in multiple languages. The translation can often be used in lieu of hand-labeling. For example, consider the polysemous word *drug*, which has two major senses: (1) a medical drug, and (2), an illicit drug. We can collect a number of sense (2) examples by extracting instances that are translated as *drogue*. In this way, we have been able to acquire considerable amounts of testing and training material for study of quantitative methods.

This testing and training material has enabled us to develop quantitative disambiguation methods that achieve 92 percent accuracy in discriminating between two senses corresponding to different topics. The relatively large amount of training materials has allowed us to study the methods for discriminating classes of examples, and to optimize the methods for this problem. We have used the class of Bayesian decision models used previously in author identification work. We find that the width of context that should be considered for topic identification is much greater than has been considered previously for sense discrimination models, and that as few as ten examples can be quite useful. The results obtained with these models seem to be substantially better than previously reported results, although comparisons are difficult.

We have also suggested several ways to overcome the bottleneck of quantitative methods: the need for examples whose sense is known. None of the methods does away with human involvement, but each can reduce the work required. They could form the basis of the part of a lexicographer's workbench devoted to determining sense differences that correspond to topic differences.

References

1. Bar-Hillel (1960), ‘‘Automatic Translation of Languages,’’ in *Advances in Computers*, Donald Booth and R. E. Meagher, eds., Academic, New York.
2. Black, Ezra (1987), *Towards Computational Discrimination of English Word Senses*, Ph. D. thesis, City University of New York.
3. Black, Ezra (1988), ‘‘An Experiment in Computational Discrimination of English Word Senses,’’ *IBM Journal of Research and Development*, v 32, pp 185-194.
4. Brown, Peter, Stephen Della Pietra, Vincent Della Pietra, and Robert Mercer (1991), ‘‘Word Sense Disambiguation using Statistical Methods,’’ *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp 264-270.
5. Brown, Peter, Jennifer Lai, and Robert Mercer (1991) ‘‘Aligning Sentences in Parallel Corpora,’’ *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp 169-176.
6. Choueka, Yaacov, and Serge Lusignam (1985), ‘‘Disambiguation by Short Contexts,’’ *Computers and the Humanities*, v 19. pp. 147-158.
7. Church, Kenneth (1989), ‘‘A Stochastic Parts Program an Noun Phrase Parser for Unrestricted Text,’’ *Proceeding, IEEE International Conference on Acoustics, Speech and Signal Processing*, Glasgow.
8. Cruse, D. A. (1986), *Lexical Semantics*, Cambridge University Press, Cambridge, England.
9. Dagan, Ido, Alon Itai, and Ulrike Schwall (1991), ‘‘Two Languages are more Informative than One,’’ *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp 130-137.
10. Fillmore, Charles, and Sue Atkins, ‘‘Word Meaning: Starting where MRD’s Stop,’’ invited talk at the 29th Annual Meeting of the Association for Computational Linguistics.
11. Granger, Richard (1977), ‘‘*FOUL-UP* A program that figures out meanings of words from context,’’ *IJCAII-77*, pp. 172-178.
12. Hearst, Marti (1991) ??? Waterloo Conference.
13. Hirschman, Lynette (1986), ‘‘Discovering Sublanguage Discovery,’’ in *Analyzing Language in Restricted Domains*, Ralph Grishman and Richard Kittredge, eds., Lawrence Erlbaum, Hillsdale, New Jersey.
14. Hirst, Graeme (???) ‘‘Semantic Interpretation and Ambiguity,’’
15. Isabelle, P. (1984) ‘‘Machine Translation at the TAUM Group,’’ in King, M. (ed.) *Machine Translation Today: The State of the Art*, Edinburgh University Press.
16. Jackson, Howard (1988) *Words and their Meaning*, Longman, London.
17. Jacobs, Paul, George Krupka, Susan McRoy, Lisa Rau, Norman Sondheimer, and Uri Zernik (1990), ‘‘Generic Text Processing: A Progress Report,’’ *Proceedings DARPA Speech and Natural Language Workshop*, pp. 359-364.
18. Kaplan, Abraham (1950), ‘‘An Experimental Study of Ambiguity in Context,’’ cited in *Mechanical Translation*, v. 1, nos. 1-3.
19. Kelly, Edward, and Phillip Stone (1975), *Computer Recognition of English Word Senses*, North-Holland, Amsterdam.
20. Kucera, H., and W. Francis (1967), *Computational Analysis of Present-day American English*, Brown University Press, Providence.

21. Lesk, Michael (1986), ‘‘Automatic Sense Disambiguation: How to tell a Pine Cone from an Ice Cream Cone,’’ *Proceeding of the 1986 SIGDOC Conference*, Association for Computing Machinery, New York.

22. Longman Group Limited, eds. (1978), *Longman Dictionary of Contemporary English*, Longman, Burnt Mill, England.

23. Masterson, Margaret (1967), ‘‘Mechanical Pidgin Translation,’’ in *Machine Translation*, Donald Booth, ed., Wiley, 1967.

24. Mosteller, Fredrick, and David Wallace (1964) *Inference and Disputed Authorship: The Federalist*, Addison-Wesley, Reading, Massachusetts.

25. Quine, W. v. O. (1960), *Word and Object*, MIT Press, Cambridge.

26. Reiger, Charles (1977), ‘‘Viewing Parsing as Word Sense Discrimination,’’ in *A Survey of Linguistic Science*, W. Dingall, ed., Greylock.

27. Sinclair, J., Hanks, P., Fox, G., Moon, R., Stock, P. et al. (eds.) (1987) *Collins Cobuild English Language Dictionary*, Collins, London and Glasgow.

28. Small, Steven (198X), ‘‘Parsing and Comprehending with Word Experts (A Theory and its Realization),’’ in WHERE???

29. Stone, Phillip, D. C. Dunphy, M. S. Smith, and D. M. Ogilvie (1966), *The General Inquirer: A Computer Approach to Content Analysis*, MIT Press, Cambridge.

30. Walker, Donald (1987), ‘‘Knowledge Resource Tools for Accessing Large Text Files,’’ in *Machine Translation: Theoretical and Methodological Issues*, Sergei Nirenberg, ed., Cambridge University Press, Cambridge, England.

31. Weinreich, U. (1980), *On Semantics*, University of Pennsylvania Press, Philadelphia.

32. Weiss, Stephen (1973), ‘‘Learning to Disambiguate,’’ *Information Storage and Retrival*, v. 9, pp 33-41.

33. Yngve, Victor (1955), ‘‘Syntax and the Prolem of Multiple Meaning,’’ in *Machine Translation of Languages*, William Locke and Donald Booth, eds., Wiley, New York.

34. Zernik, Uri (1990) ‘‘Tagging Word Senses in Corpus: The Needle in the Haystack Revisited,’’ WHERE???

Appendix

```
double
  loggam(z) double z; {
  if(z<30) return(lgamma(z));
  return(0.9189385+(z-.5)*log(z)- z + log(1+.08333333/z)); }

void
  train_one_model(n_terms,       approximate_zeros,       global_probs,
local_counts,N,V,p_wild)
 double n_terms, *global_probs, *local_counts, N, V, p_wild;
 int approximate_zeros;

{int i, term;
 double fd, sumpc, sumft, fdt, pdt, pt, ps1, pc, renorm, unseen,  w1,
w2, A, n, a;
 double f0, f1, p0, p1, k0, k1, lgsum, lgfix, lgA, lgNA, p, lam;

   f0 = .5;
   f1 = .5;

 n = n_terms;
 p = p_wild;
 sumpc = sumft = 0;
 lgfix = loggam((double) (N+1))+loggam(n+1)-loggam(N+n+1)+ 2*loggam(f1)
- loggam(2*f1);
 for(i=0; i<V; i++) {
   local[i] = local_counts[i];      /*input: word counts over document*/
   a = (double) local_counts[i];
   if(approximate_zeros && a == 0) continue;
   pdt = a/n;                 /*prob of term in doc*/
   pt=global_probs[i];
   A = pt*N;                  /*global frequency*/
   lgA = loggam(A+a+1-f0)-loggam(A+1-f0)-loggam(a+1-f1);
   lgNA = loggam(N-A+n-a+1-f0)-loggam(N-A+1-f0)-loggam(n -a+1-f1);
   lgsum = lgA + lgNA  + lgfix;
   p1 = (a+1-f1)/(n+2*(1-f1));
   p0 = (A +a +1-f0)/(N +n +2*(1-f0));
   k1 = 1;
   k0 = exp(lgsum);
   lam = (p*k1)/( (1-p)*k0 + p*k1);
   pc= (1-lam)*p0 +lam*p1;    /*the raw result*/
   fprintf(stdout,           "i: %d  lgfix: %g   lgA: %8g    lgNA:   %8g
lgsum:   %8g   k0:   %8g   lam:  %8g  p0:  %8g  p1  %8g0,
i,lgfix,lgA,lgNA,lgsum,k0,lam,p0,p1);
   fprintf(stdout, "   N= %8g A= %8g   n= %8g a= %8g0,N,A,n,a);
   local[i] = pc;
   sumpc += pc;
   sumft += pt;
 }
 renorm = 1/(sumpc+(1-sumft));
 unseen = log( 1/(sumpc+(1-sumft)));
 fprintf(stdout,
         "bayesian prior prob= %g n = %g; unseen= %g; renorm= %g; sumpg
```

```
= %g; sumpc = %g0,
         p, n, unseen, renorm, sumft, sumpc);
  for(i = 0; i < V; i++) {
   if(!approximate_zeros || local[i]>0){
     local[i] =  local[i] * renorm;
     if(approximate_zeros && fabs((double)local[i]) < .01) local[i] = 0;
   }
   else local[i] = global_probs[i]*renorm;
 } }
```