

The Interaction of Knowledge Sources in Word Sense Disambiguation

Mark Stevenson*
University of Sheffield

Yorick Wilks†
University of Sheffield

Word sense disambiguation (WSD) is a computational linguistics task likely to benefit from the tradition of combining different knowledge sources in artificial intelligence research. An important step in the exploration of this hypothesis is to determine which linguistic knowledge sources are most useful and whether their combination leads to improved results. We present a sense tagger which uses several knowledge sources. Tested accuracy exceeds 94% on our evaluation corpus.

Our system attempts to disambiguate all content words in running text rather than limiting itself to treating a restricted vocabulary of words. It is argued that this approach is more likely to assist the creation of practical systems.

1 Introduction

Word sense disambiguation (WSD) is a problem long recognised in computational linguistics (Yngve, 1955) and there has been a recent resurgence of interest with a special issue of Computational Linguistics devoted to the issue (Ide and Véronis, 1998). Despite the recent maturing of WSD within computational linguistics, there is still a considerable diversity of methods employed by researchers, as there is over the precise statement of the problems to be tackled. The SENSEVAL evaluation framework (Kilgarriff, 1998) was a DARPA-style competition designed to bring some conformity to the field of WSD although it has yet to achieve that aim completely. The main sources of divergence are the choice of computational paradigm, the proportion of text words disambiguated, the granularity of the meanings assigned to them and the knowledge sources used. We will discuss each in turn.

Resnik and Yarowsky (1997) noted that, for the most part, part-of-speech tagging is tackled using the noisy channel model, although transformation rules and grammatico-statistical methods have also had some success. There has been no consensus about WSD where a wide variety of approaches have been employed, although current approaches are moving towards machine learning methods (Yarowsky, 1995) (Rigau, Atserias, and Agirre, 1997) and combinations of classifiers (McRoy, 1992). This paper reports a WSD system employing elements of both approaches.

Another source of difference is the proportion of the vocabulary disambiguated. Some researchers have concentrated on producing WSD systems that base results on a limited number of words, for example Yarowsky (1995) and Schütze (1992) who quoted results for 12 words and a second group, such as Leacock, Towell, and Voorhees (1993) and Bruce and Wiebe (1994) who gave results for just one, namely *interest*. But limiting the

* Department of Computer Science, 211 Regent Court, Portobello Street, Sheffield S1 4DP, UK.
Email: marks@dcs.shef.ac.uk

† Department of Computer Science, 211 Regent Court, Portobello Street, Sheffield S1 4DP, UK.
Email: yorick@dcs.shef.ac.uk

vocabulary on which a system is evaluated can have two serious drawbacks. First, the words used were not chosen by frequency-based sampling techniques and so we have no way of knowing whether or not they are special cases, a point emphasised by Kilgarriff (1997) and, secondly, there is no guarantee that the techniques employed will be applicable when a larger vocabulary is tackled. It can be argued, however, that mark-up is easier when subjects have very few decisions to make, as opposed to one for each content word.

Among the researchers mentioned above, one must distinguish between supervised approaches that are inherently limited in performance to the word or words over which they evaluate because of limited training data and, on the other hand, approaches whose unsupervised learning methodology are only applied to small numbers of words for evaluation, but which could, in principle have been used to tag all content words in a text. Others, such as Harley and Glennon (1997) and ourselves Wilks and Stevenson (1998a)(1998b)(Stevenson and Wilks, 1999), have concentrated on approaches that disambiguate all content words.¹ In addition to avoiding the problems inherent in restricted vocabulary systems, wide coverage systems are more likely to be useful for NLP applications, as discussed by Wilks et al. (1990).

A third difference concerns the granularity of WSD attempted, which one can illustrate in terms of the homograph-sense distinction in LDOCE (see Section 3.1). Like Cowie, Guthrie, and Guthrie (1992) we shall give results at both levels, but it is worth pointing out that the targets of, say, work using translation equivalents (e.g. (Brown et al., 1991)(Gale, Church, and Yarowsky, 1992c), and see Section 2.3) and Roget categories (Yarowsky, 1992)(Masterman, 1957) correspond broadly to the wider, homograph, distinctions.

In this paper we attempt to show that the high level of results more typical of systems trained on many instances of a restricted vocabulary can also be obtained by large vocabulary ones, and that the best results are to be obtained from an optimisation of a combination of types of lexical knowledge (see Section 2).

1.1 Lexical Knowledge and WSD

Syntactic, semantic and pragmatic information are all potentially useful for WSD, as can be demonstrated by considering the following sentences:

- (1) John did not feel *well*.
- (2) John tripped near the *well*.
- (3) The *bat* slept.
- (4) He bought a *bat* from the sports shop.

The first two sentences contain the ambiguous word *well*; as an adjective in 1 where it is used in its ‘state of health’ sense, and as a noun in 2, meaning ‘water supply’. Since the two usages have different parts of speech they can be disambiguated by this syntactic property.

Sentence 3 contains the word *bat*, whose nominal readings are ambiguous between the meanings ‘creature’ and ‘sports equipment’. Part of speech information cannot disambiguate the senses since both are nominal usages. However, this sentence can be disambiguated using semantic information, such as preference restrictions. The verb *sleep*

¹ In this paper we define content words as nouns, verbs, adjectives and adverbs, although others have included other part of speech categories (Hirst, 1995).

prefers an animate subject and only the ‘creature’ sense of *bat* is animate. So sentence 3 can be effectively disambiguated by its semantic behaviour but not by its syntax.

A preference restriction will not disambiguate Sentence 4 since the direct object preference will be at least as general as physical object, and any restriction on the direct object slot of the verb *sell* would cover both senses. The sentence can be disambiguated on pragmatic grounds because it is far more likely that sports equipment will be bought in a sports shop. Thus pragmatic information can be used to disambiguate *bat* to its sports equipment sense.

Each of these knowledge sources has been used for WSD and in Section 3 we describe a method which performs rough-grained disambiguation using part of speech information. Wilks (1975) describes a system which performs WSD using semantic information in the form of preference restrictions. Lesk (1986) also used semantic information for WSD in the form of textual definitions from dictionaries. Pragmatic information was used by Yarowsky (1992) whose approach relied upon statistical models of categories from Roget’s Thesaurus, a resource that had been used in much earlier approaches to WSD such as Masterman (1957).

The remainder of this paper is organised as follows: Section 2 reviews some systems which have combined knowledge sources for WSD. In Section 3 we discuss the relationship between semantic disambiguation and part of speech tagging, reporting an experiment which quantifies the connection. A general WSD system is presented in Section 4. In Section 5 we explain the strategy used to evaluate this system and report the results in Section 6.

2 Background

A comprehensive review of WSD is beyond the scope of this paper but may be found in Ide and Véronis (1998), and Combining knowledge sources for WSD is not a new idea; in this section we will review some of the systems which have tried to do that.

2.1 McRoy’s System

An early attempt to carry out coarse-grained WSD by combining knowledge sources was carried out by McRoy (1992). Her work was carried out without the use of machine readable dictionaries (MRD) leading to the manual creation of the complex set of lexicons this system requires. There was a lexicon of 8,775 unique roots, a hierarchy of 1,000 concepts and a set of 1,400 collocational patterns. The collocational patterns are automatically extracted from a corpus of text in the same domain as the text being disambiguated and senses manually assigned to each. If the collocation occurs in the text being disambiguated then it is assumed that the words it contains are being used in the same senses as were assigned manually.

Disambiguation makes use of several knowledge sources: frequency information, syntactic tags, morphological information, semantic context (clusters), collocations and word associations, role-related expectations and selectional restrictions. The knowledge sources are combined by adding their results. Each knowledge source assigns a (possibly negative) numeric value to each of the possible senses. The numerical value depends upon the type of knowledge source. Some knowledge sources have only two possible values, for example the frequency information has one value for frequent senses and another for infrequent ones. The numerical values assigned for each were determined manually. The selectional restrictions knowledge source assigns scores in the range -10 to +10, with higher scores being assigned to senses that are more specific (according to the concept hierarchy). Disambiguation is carried out by summing the scores from each knowledge sources for all candidate senses and choosing the one with the highest overall score.

In a sample of 25,000 words from the Wall Street Journal the system covered 98% of non-proper noun, non-abbreviated word-occurrences demonstrating the impressive coverage of the hand-crafted lexicons. No quantitative evaluation of the disambiguation quality was carried out due to the difficulty in obtaining annotated test data, a problem made more acute by the use of a bespoke lexicon. In addition, comparison of system output against manually annotated text had yet to become a standard evaluation strategy in WSD research.

2.2 The Cambridge Language Survey System

The *Cambridge International Dictionary of English* (CIDE) (Procter, 1995) is a learners' dictionary which consists of definitions written using a 2,000 word controlled vocabulary. (This lexicon is similar to LDOCE which we use for experiments presented later in this paper, it is described in Section 3.1.) The senses in CIDE are grouped by guidewords, similar to homographs in LDOCE. It was produced using a large corpus of English created by the Cambridge Language Survey (CLS).

The CLS also produced a semantic tagger (Harley and Glennon, 1997), a commercial product which tags words in text with senses from their MRD. The tagger consists of four sub-taggers running in parallel with their results being combined after all have run. The first tagger uses collocations derived from the CIDE example sentences. The second examines the subject codes for all words in a particular sentence and the number of matches with other words calculated. A part-of-speech tagger produced in-house by CUP is run over the text and high scores assigned to senses which agree with the syntactic tag assigned. Finally, the selectional restrictions of verbs and adjectives are examined. The results of these processes are combined using a simple weighting scheme (similar to McRoy's, see Section 2.1), which was inspired by those used in computer chess programs, and which assigns each sub-process a weight in the range -100 to +100 before summing. Unlike McRoy this approach does not consider the specificity of a knowledge source in a particular instance but assigns an overall weight to each which is always applied.

Harley and Glennon report 78% correct tagging of all content words at the CIDE guideword level, which they equate to the LDOCE sense level, and 73% at the sub-sense level as compared to a hand-tagged corpus of 4,000 words.

2.3 Machine Learning applied to WSD

An early application of machine learning to the WSD problem was carried out by Brown et al. (1991). Several different disambiguation cues such as first noun to the left/right and second word to the left/right were extracted from parallel text. Translation differences were used to define the senses as this approach was used in an English-French machine translation system. The parallel text effectively provided supervised training examples for this algorithm. Nadas et al. (1991) used the flip-flop algorithm to decide which of the cues was most important for each word by maximising mutual information scores between words. Yarowsky (1996) used an extremely rich features set by expanding this set with syntactic relations such as subject-verb, verb-object and adjective-noun relations, part-of-speech n-grams and others. The approach was based on the hypothesis that words exhibited "one sense per collocation" (Yarowsky, 1993). A large corpus was examined to compute the probability of a particular collocate occurring with a certain sense and the discriminatory power of each calculated using the log-likelihood ratio. These ratios were used to create a decision list with the most discriminating collocations being preferred. This approach has the benefit that it does not combine the probabilities of these highly non-independent knowledge sources.

Yarowsky (1993) also examined the discriminatory power of the individual knowledge sources. It was found that each collocation indicated a particular sense with a very high

degree of reliability with the most successful, the first word to the left of a noun, achieving 99% precision. Yet collocates have limited applicability; although precise they can only be applied to a limited number of tokens. Yarowsky (1995) largely dealt with this problem by producing an unsupervised learning algorithm which generates probabilistic decision list models of word senses from seed collocates and achieves 97% correct disambiguation. In these experiments Yarowsky deals exclusively with binary sense distinctions and evaluates his highly effective algorithms on small samples of word tokens.

Ng and Lee (1996) explored an approach to WSD in which a word is assigned the sense of the most similar example already seen. They describe this approach as “exemplar-based learning” although it is also known as k-nearest neighbour learning. Their system is known as LEXAS (LEXical Ambiguity-resolving System), a supervised learning approach which requires disambiguated training text. LEXAS was based on PEBLS, a publically available exemplar-based learning algorithm.

A set of features is extracted from disambiguated example sentences, including part of speech information, morphological form, surrounding words, local collocates and words in verb-object syntactic relations. When a new, untagged, usage is encountered it is compared with each of the training examples and the distance from each is calculated using a metric adopted from Cost and Salzberg (1993). This is calculated as the sum of the differences between each pair of features in the two vectors. The differences between two values v_1 and v_2 is calculated according to (5) where $C_{1,i}$ represents the number of training examples with value v_1 that is classified with sense i in the training corpus, and C_1 the number with value v_1 in any sense. $C_{2,i}$ and C_2 denote similar values and n the total number of senses for the word under consideration. The sense of the example with the minimum distance from the untagged usage is chosen, if there is more than one with the same distance, one is chosen at random to break the tie.

$$\delta(v_1, v_2) = \sum_{i=1}^n \left| \frac{C_{1,i}}{C_1} - \frac{C_{2,i}}{C_2} \right| \quad (5)$$

Ng and Lee tested LEXAS on two separate data sets: one used previously in WSD research, the other a new, manually tagged, corpus. The common data set was the *interest* corpus constructed by Bruce and Wiebe (1994) consisting of 2,639 sentences from the Wall Street Journal, each containing an occurrence of the noun *interest*. Each occurrence is tagged with one of its six possible senses from LDOCE. Evaluation is carried out through 100 random trials, each trained on 1,769 sentences and tested on the 600 remaining. The average accuracy was 87.4%, significantly higher than the figure of 78% reported by Bruce and Wiebe.

Further evaluation was carried out on a larger data set constructed by Ng and Lee. This consisted of 192,800 occurrences of the 121 nouns and 70 verbs which are “the most frequently occurring and ambiguous words in English” (Ng and Lee, 1996, p. 44). The corpus was made up from the Brown Corpus (Kučera and Francis, 1967) and the Wall Street Journal Corpus and was tagged with the correct senses from WordNet by university undergraduates specialising in linguistics. Before training, two subsets of the corpus were put aside as test sets: the first (BC50) contains 7,119 occurrences of the ambiguous words from the Brown Corpus while the second (WSJ6) contained 14,139 from the Wall Street Journal Corpus. LEXAS correctly disambiguated 54% of words in BC50 and 68.6% in WSJ6. Ng and Lee point out that both results are higher than choosing the first or most frequent sense in each of the corpora. The authors attribute the lower performance on the Brown Corpus to the wider variety of text types it contains.

Ng and Lee attempted to determine the relative contribution of each knowledge source. This was carried out by re-running the data from the *interest* corpus through the

learning algorithm, this time removing all but one set of features. The results are shown in Table 1. They found that the local collocations were the most useful knowledge source in their system. However, it must be remembered that this experiment was carried out on a data set consisting of a single word and may, therefore, not be generalisable.

Table 1
Relative Contribution of Knowledge Sources in LEXAS

Knowledge Source	Accuracy
Collocations	80.2%
PoS and Morphology	77.2%
Surrounding words	62.0%
Verb-object	43.5%

2.4 Discussion

This review has been extremely brief and has not covered large areas of research into WSD. For example, we have not discussed connectionist approaches, as used by Waltz and Pollack (1985), Véronis and Ide (1990), Hirst (1987) and Cottrell (1984). However, we have attempted to discuss some of the approaches to combining diverse types of linguistic knowledge for WSD and have concentrated on those which are related to the techniques used in our own disambiguation system.

Of central interest to our research is the relative contribution of the various knowledge sources which have been applied to the WSD problem. Both Ng and Lee (1996) and Yarowsky (1993) reported some results in the area. However, Ng and Lee reported results for only a single word and Yarowsky considers only words with two possible senses. This paper is an attempt to increase the scope of this research by discussing a disambiguation algorithm which operates over all content words and combines a varied set of linguistic knowledge sources. In addition, we examine the relative effect of each knowledge source to gauge which are the most important and under what circumstances.

We first report an in-depth study of a particular knowledge source, namely part-of-speech tags.

3 Part of Speech and Word Senses

3.1 LDOCE

The experiments described in this section use the *Longman Dictionary of Contemporary English* (LDOCE) (Procter, 1978). LDOCE is a learners' dictionary, designed for students of English, containing roughly 36,000 word types. LDOCE was innovative in its use of a defining vocabulary of 2,000 words with which the definitions were written. If a learner of English could master this small core then, it was assumed, they could understand every entry in the dictionary.

In LDOCE, the senses for each word type are grouped into *homographs*: sets of senses with related meanings. For example, one of the homographs of *bank* means roughly 'things piled up', with different senses distinguishing exactly what is piled up (see Figure 1). If the senses are sufficiently close together in meaning there will be only one homograph for that word which we then call *monohomographic*. However, if the senses are far enough apart, as in the *bank* case, they will be grouped into separate homographs, which we call *polyhomographic*.

As can be seen from the example entry, each LDOCE homograph includes informa-

bank¹ *n* **1** land along the side of a river, lake, etc. **2** earth which is heaped up in a field or a garden, often making a border or division **3** a mass of snow, mud, clouds, etc.: *The banks of dark cloud promised a heavy storm* **4** a slope made at bends in a road or race-track, so that they are safer for cars to go round **5** SANDBANK: *The Dogger Bank in the North Sea can be dangerous for ships*

bank² *v* [IØ] (of a car or aircraft) to move with one side higher than the other, esp. when making a turn – see also BANK UP

bank³ *n* **1** a row, esp. of OARS in an ancient boat or KEYS on a TYPEWRITER

bank⁴ *n* **1** a place where money is kept and paid out on demand, and where related activities go on – see picture at STREET **2** (*usu. in comb.*) a place where something is held ready for use, esp. ORGANIC product of human origin for medical use: *Hospital bloodbanks have saved many lives* **3** (a person who keeps) a supply of money or pieces for payment or use in a game of chance **4** **break the bank** to win all the money that the BANK⁴(3) has in a game of chance

bank⁵ *v* **1**[T1] to put or keep (money) in a bank **2**[L9, esp. *with*] to keep one's money (esp. in the stated bank): *Where do you bank?*

Figure 1

The entry for *bank* in LDOCE (slightly simplified for clarity)

tion about the part of speech with which the homograph is marked and which applies to each of the senses within that homograph. However, about 2% of word types in LDOCE contain a homograph which has some combination of part of speech (e.g. noun OR verb) associated with each of its senses.

Although the granularity of the distinction between homographs in LDOCE is rather coarse-grained, they are, as we noted at the beginning of this paper, an appropriate level for many practical computational linguistic applications. For example, *bank* in the sense of ‘financial institution’ translates to *banque* in French but when used in the ‘edge of river’ sense it translates as *bord*. This level of semantic disambiguation is frequently sufficient for choosing the correct target word in an English-to-French Machine Translation system and is at a similar level of granularity to the sense distinctions explored by other researchers in WSD, for example Brown et al. (1991), Yarowsky (1996) and McRoy (1992) (see Section 2).

3.2 Using Part of Speech Information to Resolve Senses

We began by examining the potential usefulness of part of speech information for sense resolution. It was found that 34% of the content word types in LDOCE were polysemous and 12% polyhomographic. (Polyhomographic words are necessarily polysemous since each homograph is a non-empty set of senses.) If we assume that the part of speech of each polyhomographic word in context is known then 88% of them (word types that is) would be disambiguated to the homograph level. (In other words, 88% do not have two homographs with the same part of speech.) Some words will be disambiguated to the homograph level if they are used in a certain part of speech but not others. For example, *beam* has 3 homographs in LDOCE, the first two are marked as nouns while the third is marked as verb. This word would be disambiguated if it is used as a verb but not if used as a noun. If we assume that every word of this type is assigned a part of speech which disambiguates it (i.e. Verb in the case of *beam*) then an additional 7% of words in LDOCE

could, potentially, be disambiguated. Therefore, up to 95% of word types in LDOCE can be disambiguated to the homograph level by part of speech information alone. However, these figures do not take into account either errors in part-of-speech tagging or the corpus distribution of tokens since each word type is counted exactly once.

The next stage in our analysis was to attempt to disambiguate some texts using the information obtained from part-of-speech tags. We took five articles from the *Wall Street Journal*, containing 391 polyhomographic content words. These articles were manually tagged with the most appropriate LDOCE homograph by one of the authors. The texts were then part-of-speech tagged using Brill's transformational-based learning tagger (Brill, 1995). The tags assigned by the Brill tagger were manually mapped onto the simpler part-of-speech tag set used in LDOCE.² Part-of-speech tags cannot always identify a single homograph (if the word has more than one homograph with a particular part of speech), and in such cases we chose the first sense listed in LDOCE since this is the one which occurs most frequently.³

It was found that 87.4% were assigned the correct homograph. A baseline for this task can be calculated by computing the number of tokens which would be correctly disambiguated if the first homograph for each was chosen regardless of part of speech. 78% of polyhomographic tokens were correctly disambiguated this way using this approach.

These results show there is a clear advantage to be gained by using the very simple part-of-speech based method described compared with simply choosing the first homograph (over 42% reduction in error rate). However, we felt that it would be useful to carry out some further analysis of the data. To do this it is useful to divide the polyhomographic words into four classes, all based on the assumption that a part-of-speech tagger has been run over the text and that homographs which do not correspond to the grammatical category assigned have been removed.

Full disambiguation (by part of speech) If only a single homograph with the correct part of speech remains, that word has been fully disambiguated by the tagger.

Partial disambiguation (by part of speech) If there is more than one possible homograph with the correct part of speech but some have been removed from consideration we then say that it has been partially disambiguated by part of speech.

No disambiguation (by part of speech) If all the homographs of a word have the same part of speech, which is then assigned by the tagger, then none can be removed and no disambiguation carried out.

Part of speech error It is possible for the part of speech tagger to assign an incorrect part of speech leading to the correct homograph being removed from consideration. It is worth mentioning that this situation has two possible outcomes; first, some homographs, with incorrect parts of speech, remain or, secondly, all homographs may be removed from consideration.

² The Brill tagger uses the 48 tag set from the Penn Tree Bank (Marcus, Santorini, and Marcinkiewicz, 1993), while LDOCE uses a set of 17 more general tags. Brill's tagger has a reported error rate of around 3%, although we found that mapping the Penn TreeBank tags used by Brill onto the simpler LDOCE tag set led to a lower error rate.

³ In the 3rd Edition of LDOCE the publishers claim that the senses are indeed ordered by frequency, although they make no such claim of the 1st Edition, as used here. However, Guo (1989) found evidence that there is a correspondence between the order in which sense are listed and the frequency of occurrence in the 1st Edition.

Table 2

Examples of four word types introduced in Section 3.2. The leftmost column indicates the full set of homographs for the example words, with upper case indicating the correct homograph. The remaining columns show (respectively) the part-of-speech assigned by the tagger, the resulting set of senses after filtering and the type of the word.

All Homographs	PoS Tag	After tagging	Word type
N, v, v	n	N	Full disambiguation
n, adj, V	v	V	Full disambiguation
n, V, v	v	V, v	Partial disambiguation
n, N, v	n	n, N	Partial disambiguation
N, n	n	N, n	No disambiguation
v, V	v	v, V	No disambiguation
N, v, v	v	v v	PoS error
N, v, v	adj	N, v, v	PoS error

Table 3

Error analysis for the experiment on WSD by part of speech alone

Word Type	Count	Correctly disambiguated by:	
		Baseline method	PoS method
Fully disambiguation	297	268 (90%)	297 (100%)
Partial disambiguation	58	22 (38%)	32 (55%)
No disambiguation	23	10 (43%)	10 (43%)
Part of speech error	13	5 (38%)	3 (23%)
All polyhomographic	391	305 (78%)	342 (87%)

In Table 3 we show the number of words in our five articles which fall into each of the four categories in the column labelled “Count”. The relative performance of the baseline method (choosing first sense) compared to the reported algorithm (removing homographs using part of speech tags) are shown in the rightmost two columns. The figures in brackets indicate the percentage correctly disambiguated by each method on a per-class basis. It can be seen that the majority of the polyhomographic words (297 of 342) fall into the “Full disambiguation” category, all of which are correctly disambiguated by the method reported here. When no disambiguation is carried out then the algorithm described simply chooses the first sense and so the results are the same for both methods. The only condition under which choosing the first sense is more effective than using part of speech information is when the part-of-speech tagger makes an error, and all the homographs with the correct part of speech are removed from consideration. In most cases this means that the correct homograph cannot be chosen but, in a small number of cases, this is equivalent to choosing the most frequent since, if all possible homographs have been removed from consideration, the algorithm reverts to using the simpler heuristic of choosing the word’s first homograph.⁴

Although this result may seem intuitively obvious there have, we believe, been no other attempts to quantify the benefit to be gained from the application of a part-of-speech tagger in WSD (see (Wilks and Stevenson, 1998a)). The method described here is effective at removing incorrect senses from consideration thereby reducing the search space if combined with other WSD methods.

In the experiments reported in this section we made use of the particular structure of LDOCE which assigns each sense to a homograph from which its part of speech information is inherited. However, there is no reason to believe that the method reported here is limited to lexicons with this structure. In fact this approach can be applied to any lexicon which assigns part of speech information to senses, although it would not always be possible to evaluate at the homograph level, as we do here.

In the remainder of this paper we go on to describe a sense tagger that assigns senses from LDOCE using a combination of classifiers. The set of senses which is considered by the classifiers is first filtered using part-of-speech tags.

4 A Sense Tagger which Combines Knowledge Sources

We adopt a framework in which different knowledge sources are applied as separate modules. One type of module can be used to remove senses from consideration when a knowledge source identifies them as unlikely in context i.e. a *filter*. Another type can be used when a knowledge source provides evidence for a sense but cannot identify it confidently and we call these *partial taggers* (in the spirit of McCarthy’s notion of “partial information” (REF)). The choice of whether to apply a knowledge source as either a filter or partial tagger depends on whether it is likely to rule out correct senses. If a knowledge source is unlikely to reject the correct sense then it can be safely implemented as a filter, otherwise implementation as a partial tagger would be more appropriate. In addition, it is necessary to represent the context of ambiguous words so that this information can be used in the disambiguation process. In the system described here these modules are referred to as *feature extractors*.

Our sense tagger is implemented within this modular architecture, one where each module is a filter, partial tagger or feature extractor. The architecture of the system is represented in Figure 2. This system currently incorporates a single filter (**part-of-speech**

⁴ An example of this situation is shown in the bottom row of Table 2.

filter), three partial taggers (simulated annealing, subject codes, selectional restrictions) and a single feature extractor (collocation extractor).

4.1 Preprocessing

Before the filters or partial taggers are applied, the text is tokenised, lemmatised, split into sentences and part-of-speech tagged, again using Brill's tagger. A named entity identifier is then run over the text to mark and categorise proper names, which will provide information for the `selectional restrictions` partial tagger (see Section 4.4). These pre-processing stages are carried out by modules from Sheffield University's Information Extraction system, *LaSIE*, and are described in more detail by Gaizauskas et al. (1996).

Our system disambiguates only the content words in the text, and the part-of-speech tags are used to decide which are content words. There is no attempt to disambiguate any of the words identified as part of a named entity. These are excluded because they have already been semantically analysed through the classification added by the named entity identifier (see Section 4.4). Another reason for not attempting WSD on named entities is that when words are used as names they are not being used in any of the senses listed in a dictionary. For example, *Rose* and *May* are names but there no senses in LDOCE for this usage. It may be possible to create a dummy entry in the set of LDOCE senses indicating that the word is being used as a name but then the sense tagger would simply repeat work carried out by the named entity identifier.

4.2 Part of Speech filtering

We take the part-of-speech tags assigned by the Brill tagger and use a manually created mapping to translate these to the corresponding LDOCE grammatical category (see Section 3.2). Any senses which do not correspond to the category returned are removed from consideration. In practice, the filtering is carried out at the same time as the lexical lookup phase and the senses whose grammatical categories do not correspond to the tag assigned are never attached to the ambiguous word. There is also an option to turn off filtering so that all senses are attached regardless of the part-of-speech tag. If none of the dictionary senses for a given word agree with the part-of-speech tag then all are kept.

One could reasonably argue that removing senses is a dangerous strategy since, if the part-of-speech tagger made an error, the correct sense could be removed from consideration. However, the experiments described in Section 3.2 indicate that part of speech information is unlikely to reject the correct sense and can be safely implemented as a filter.

4.3 Optimising Dictionary Definition Overlap

Lesk (1986) proposed that WSD could be carried out using an overlap count of content words in dictionary definitions as a measure of semantic closeness. This method would tag all content words in a sentence with their senses from a dictionary which contains textual definitions. However, it was found that the computations which would be necessary to test every combination of senses, even for a sentence of modest length, was prohibitive.

The approach was made practical by Cowie, Guthrie, and Guthrie (1992)(see also (Wilks, Slator, and Guthrie, 1996)) who computed the overlap using the simulated annealing optimisation algorithm (Metropolis et al., 1953) which approximates the results of calculating all possible combinations of senses. Although simulated annealing is not guaranteed to find the global solution to an optimisation problem it has been shown to find solutions which are not significantly different from the optimal one (Press et al., 1988). Cowie et. al. used LDOCE for their implementation which correctly disambiguated 47% of words to the sense level, 72% to the homograph level compared with manually assigned senses. In their implementation the optimisation was carried out over a simple

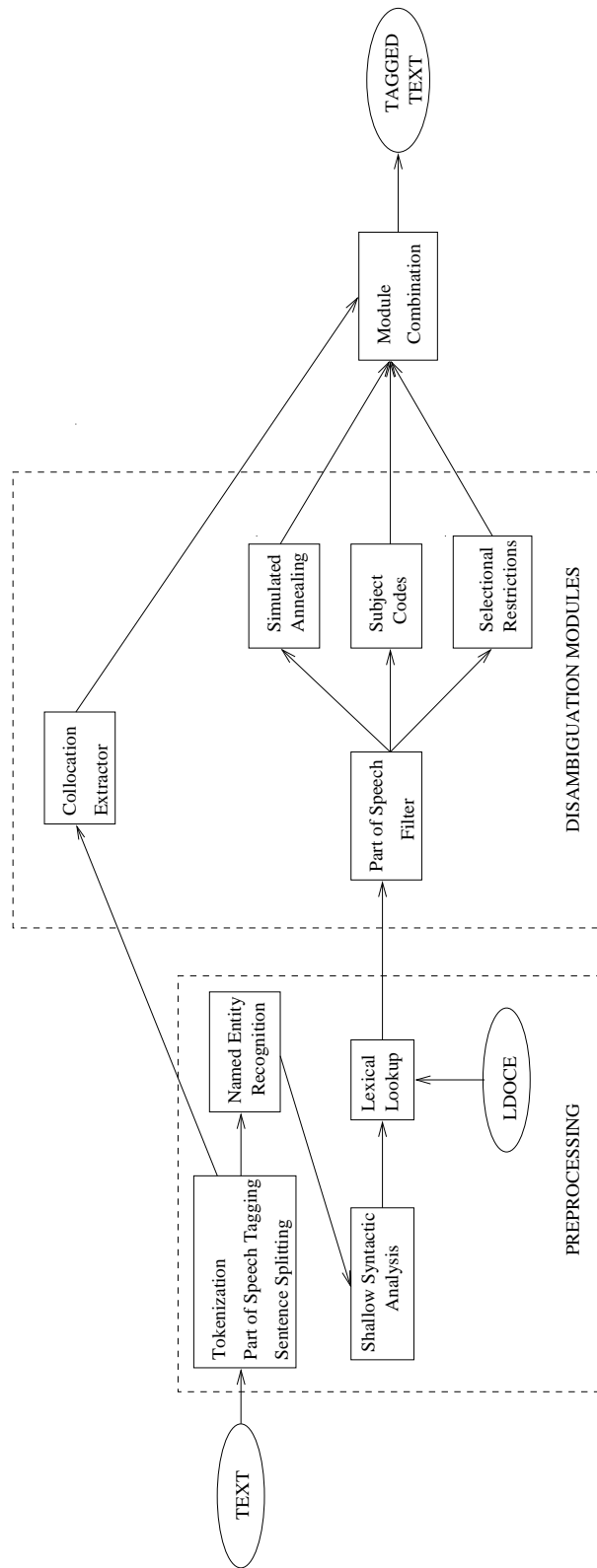


Figure 2
Sense Tagger Architecture

count of words in common between definitions, although this meant that longer definitions were preferred (since they have more words which can contribute to the overlap) and short definitions or definitions by synonym were correspondingly penalised. We addressed this problem by computing the overlap in a different way: instead of each word contributing one, we normalised its contribution by the number of words in the definition it came from. In their implementation Cowie et. al. also added pragmatic codes to the overlap computation, however we prefer to keep different knowledge sources separate and use this information in another partial tagger (see Section 4.5). The Cowie et. al. implementation returned one sense for each ambiguous word in the sentence, without any indication of the system’s confidence in its choice, but we adapted the system to return a set of suggested senses for each ambiguous word in the sentence.

4.4 Selectional Preferences

Our next partial tagger returns the set of senses for each word which are licensed by selectional preferences (in the sense of Wilks (1975)). LDOCE senses are marked with selectional restrictions expressed by 36 semantic codes not ordered in a hierarchy. However, the codes are clearly not of equal levels of generality, for example, the code H is used to represent things which are human while M represents human males. Thus for a restriction with type H, we would want to allow words with the more specific semantic class M to meet it. This can be computed if the semantic categories are formed into a hierarchy. Then all categories subsumed by another will be regarded as satisfying the restriction. Bruce and Guthrie (1992) manually identified relations between the LDOCE semantic classes, grouping the codes into small sets with roughly the same meaning and attached descriptions; for example M, K are grouped as a pair described as “human male”. The hierarchy produced is shown in Figure 3.

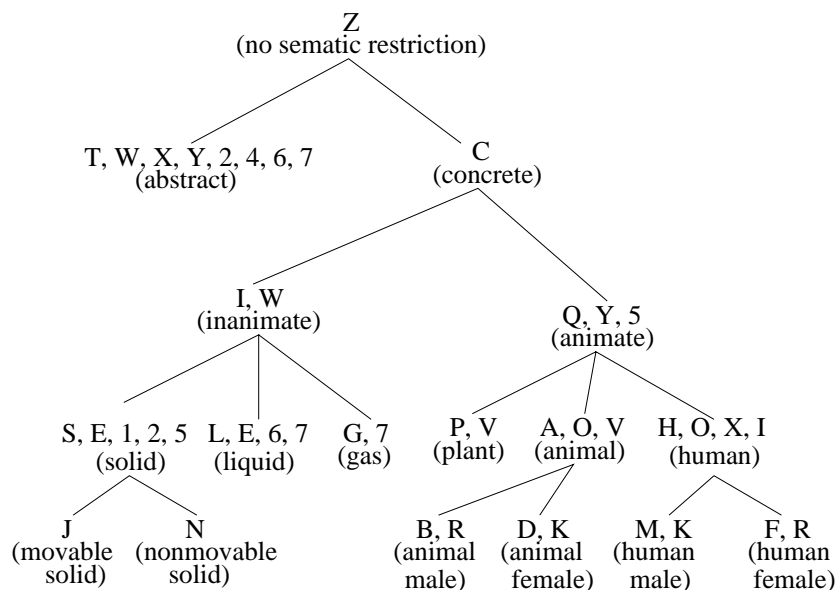


Figure 3
Bruce and Guthrie’s hierarchy of LDOCE semantic codes

The named entities identified as part of the preprocessing phase (Section 4.1) are used by this module, which requires first a mapping between the name types and LDOCE

semantic codes, and it is shown in Table 4.

Table 4

Mapping of named entities onto LDOCE semantic codes. The named entities can be mapped to any semantic code within a particular node of the hierarchy since the disambiguation algorithm treats all codes in the same node as equivalent.

Named Entity	Type	LDOCE code
PERSON		H (= Human)
ORGANISATION		T (= Abstract)
LOCATION		N (= Non-movable solid)
DATE		T (= Abstract)
TIME		T (= Abstract)
MONEY		T (= Abstract)
PERCENT		T (= Abstract)
UNKNOWN		Z (= No semantic restriction)

Any use of preferences for sense selection requires prior identification of the sentence sites of relationship. Although this was not done by syntactic methods in (Wilks, 1975) it is often easiest to think of the relationships specified in grammatical terms, as e.g. subject-verb, verb-object, adjective-noun etc. We perform this step by means of a shallow syntactic analyser (Stevenson, 1998) which finds the following grammatical relations: the subject, direct and indirect object of each verb (if any) and the noun modified by an adjective. Stevenson (1998) describes an evaluation of this system in which the relations identified were compared with those derived from Penn TreeBank parses (Marcus, Santorini, and Marcinkiewicz, 1993). It was found that the parser achieved 51% precision and 69% recall.

Our preference resolution algorithm begins by computing the set of verb senses for which there is a set of noun senses which satisfy the preferences placed on them. For each noun dominated by that verb this produces a set of senses which agree with at least one verb sense. For each noun we then check whether it is modified by an adjective. If it is we reject any senses of the adjectives which do not agree with any of the remaining noun senses. This approach is rather conservative in that it does not reject a sense unless it is impossible for it to fit into the preference pattern of the sentence.

Table 5

Toy Example

Example sentence: <i>John ran the hilly course.</i>		
Sense	Definition and <i>Example</i>	Restriction
John	proper name	type:human
ran (1)	to control an organisation <i>Gates runs Microsoft</i>	subject:human object:abstract
ran (2)	to move quickly by foot <i>Redrum ran the steeplechase</i>	subject:human object:inanimate
hilly (1)	undulating terrain <i>hilly road</i>	modifies:nonmovable solid
course (1)	route <i>race course</i>	type:nonmovable solid
course (2)	programme of study <i>physics course</i>	type:abstract

In order to explain this process more fully we provide a walk-through explanation of the procedure applied to a toy example shown in Table 5. It is assumed that the name

entity identifier has correctly identified *John* as a person and that the shallow parser has found the correct syntactic relations. In order to make this example as straightforward as possible we consider only the case in which the ambiguous words have few senses. The disambiguation process operates by considering the relations between the words in known grammatical relations and before it begins we essentially have a set of possible senses for each word related via their syntax. This situation is represented by the topmost tree in Figure 4.

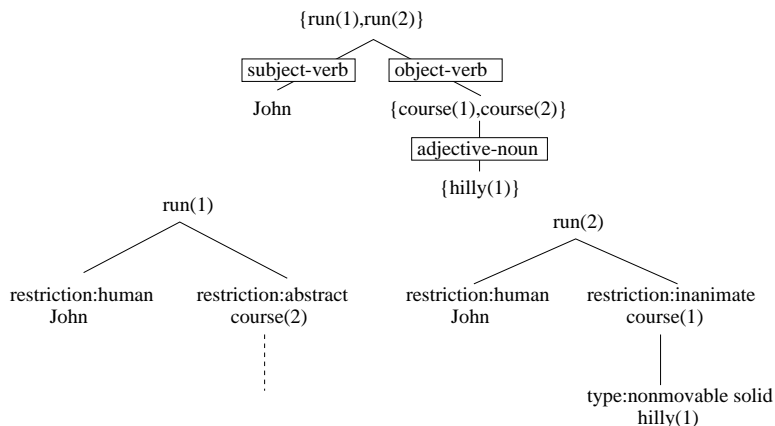


Figure 4
Restriction Resolution in Toy Example

Disambiguation is carried out by considering each verb sense in turn, beginning with *run(1)*. As *run* is being used transitively it places two restrictions on the sentence: first, the subject must satisfy the restriction **human** and the object **abstract**. In this example, *John* has been identified as a named entity and marked as **human** so the subject restriction is not broken. Note that, if the restriction was broken, then the verb sense *run(1)* would be marked as incorrect by this partial tagger and no further attempt would be made to resolve its restrictions. As this was not the case we consider the direct object slot, which places the restriction **abstract** on the noun which fills it. *course(2)* fulfils this criterion. *course* is modified by *hilly* which expects a noun of type **nonmovable solid**. However, *course(2)* is marked **abstract** which does not comply with this restriction. Therefore, assuming that *run* is being used in its second senses leads to a situation in which there is no set of senses which comply with all the restrictions placed on them and therefore *run(1)* is not the correct sense of *run* and the partial tagger marks this sense as wrong. This situation is represented by the tree on the bottom left of Figure 4. The sense *course(2)* is not rejected at this point since it may be found to be acceptable in the configuration of senses of another sense of *run*.

The algorithm now assumes that *run(2)* is the correct sense. This implies that *course(1)* is the correct sense as it complies with the **inanimate** restriction that verb sense places on the direct object. As well as complying with the restriction imposed by *run(2)*, *course(1)* also complies with the one imposed by *hilly(1)* (since **nonmovable solid** is subsumed by **inanimate**). Therefore, assuming that senses *run(2)* and *course(1)* are being used does not lead to any restrictions being broken and the algorithm marks these as correct.

Before leaving this example it is worth discussing a few points. The sense *course(2)* is marked as incorrect since there is no sense of *run* with which an interpretation of

the sentence can be constructed using *course(2)*. If there were further senses of *run* in our example, and *course(2)* was found to be suitable for those extra senses, then the algorithm would mark the second sense of *course* as correct. There is, however, no condition under which *run(1)* could be considered as correct through the consideration of further verb senses. Also, although *John* and *hilly* are not ambiguous in this example they still participate in the disambiguation process. In fact they are vital to its success as the correct senses could not have been identified without considering the restrictions placed by the adjective *hilly*.

This partial tagger returns, for all ambiguous noun, verb and adjective occurrences in the text, the set of senses which satisfy the preferences imposed on them. Adverbs do not have any selectional preferences in LDOCE and so are ignored by this partial tagger.

4.5 Subject Codes

Our final partial tagger is a re-implementation of the algorithm developed by Yarowsky (1992). This algorithm is dependent upon a categorisation of words in the lexicon into subject areas, Yarowsky used the Roget large categories. In LDOCE, primary pragmatic codes indicate the general topic of a text in which a sense is likely to be used. For example, LN means “Linguistics and Grammar” and this code is assigned to some senses of words such as “ellipsis”, “ablative”, “bilingual” and “intransitive”. Roget is a thesaurus so each entry in the lexicon belongs to one of the large categories, but over half (56%) of the senses in LDOCE are not assigned a primary code. We therefore created a dummy category, denoted by --, used to indicate a sense which is not associated with any specific subject area and this category is assigned to all senses without a primary pragmatic code. These differences between the structures of LDOCE and Roget meant that we had to adapt the original algorithm reported in Yarowsky (1992).

In Yarowsky’s implementation, the correct subject category is estimated by applying (6). The maximises the sum of a Bayesian term (the fraction on the right) over all possible subject categories (*SCat*) for the ambiguous word over the words in its context (*w*). A context of 50 words on either side of the ambiguous word is used.

$$\underset{SCat}{ARGMAX} \sum_{w \in context} \log \frac{\Pr(w|SCat) \Pr(SCat)}{\Pr(w)} \quad (6)$$

Yarowsky assumed the prior probability of each subject category to be constant, so the value $\Pr(SCat)$ has no effect on the maximisation in (6), and (7) was effectively being maximised.

$$\underset{SCat}{ARGMAX} \sum_{w \in context} \log \frac{\Pr(w|SCat)}{\Pr(w)} \quad (7)$$

By including a general pragmatic code to deal with the lack of coverage, we created an extremely skewed distribution of codes across senses and Yarowsky’s assumption that subject codes occur with equal probability is unlikely to be useful in this application. We gained a rough estimate of the probability of each subject category by determining the proportion of senses in LDOCE assigned it and applying the maximum likelihood estimate. It was found that results improved when the rough estimate of the likelihood of pragmatic codes was used. It is possible that this estimate may be improved by counting probabilities by token rather than by type, as was done here, although the problem of polysemy in the training data would have to be overcome in some way.

The algorithm relies upon the calculation of probabilities gained from corpus statistics: Yarowsky used the Grolier’s Encyclopaedia, which comprised a 10 million word corpus. Our implementation used nearly 14 million words from the non-dialogue portion

of the British National Corpus (Burnard, 1995). Yarowsky used smoothing procedures to compensate for data sparseness in the training corpus, detailed in (Gale, Church, and Yarowsky, 1992b), which we did not implement. Instead, we attempted to avoid this problem by considering only words which appeared at least 10 times in the training contexts of a particular word. A context model is created for each pragmatic code by examining 50 words either side of any word in the corpus containing a sense marked with that code. Disambiguation is carried out by examining the same 100 word context window for an ambiguous word and comparing it against the models for each of its possible categories. Further details may be found in (Yarowsky, 1992).

Yarowsky reports 92% correct disambiguation over 12 test words, with an average of three possible Roget large categories. However, LDOCE has a higher level of average ambiguity and does not contain as complete a thesaural hierarchy as Roget, so we would not expect such good results when the algorithm is adapted to LDOCE. Consequently, we implemented the approach as a partial tagger. The algorithm identifies the most likely pragmatic code and returns the set of senses which are marked with that code. In LDOCE several senses of a word may be marked with the same pragmatic code so this partial tagger may return more than one sense for an ambiguous word.

4.6 Collocation Extractor

The final disambiguation module is the only feature extractor in our system and is based on collocations. A set of 10 collocates are extracted for each ambiguous word in the text: first word to the left, first word to the right, second word to the left, second word to the right, first noun to the left, first noun to the right, first verb to the left, first verb to the right, first adjective to the left and first adjective to the right. Some of these types of collocation were also used by Brown et al. (1991) and Yarowsky (1993) (see Section 2.3). All collocates are searched for within the sentence which contains the ambiguous word. If some particular collocation does not exist for an ambiguous word, for example if it is the first or last word in a sentence, then a null value (NoColl) is stored instead. Rather than storing the surface form of the co-occurrence, morphological roots are stored instead as this allows for a smaller set of collocations, helping to cope with data sparseness. The surface form of the ambiguous word is also extracted from the text and stored. The extracted collocations and surface form combine to represent the context of each ambiguous word.

4.7 Combining Disambiguation Modules

The results from the disambiguation modules (filter, partial taggers and feature extractor) are then presented to a machine learning algorithm to combine their results. The algorithm we chose was the TiMBL memory-based learning algorithm (Daelemans et al., 1999). Memory-based learning is another name for exemplar-based learning, as employed by Ng and Lee (Section 2.3). The TiMBL algorithm has already been used for various NLP tasks including part-of-speech tagging and PP-attachment (Daelemans et al., 1996) (Zavrel, Daelemans, and Veenstra, 1997).

Like PEBLS, which formed the core of Ng and Lee’s LEXAS system, TiMBL classifies new examples by comparing them against previously seen cases. The class of the most similar example is assigned. At the heart of this approach is the distance metric $\Delta(X, Y)$ which computes the similarity between instances X and Y . This measure is calculated using the weighted overlap metric shown in (8) which calculates the total distance by computing the sum of the distance between each position in the feature vector.

$$\Delta(X, Y) = \sum_{i=1}^n w_i \delta(x_i, y_i) \quad (8)$$

where:

$$\delta(x_i, y_i) = \begin{cases} \frac{x_i - y_i}{\max_i - \min_i} & \text{if numeric, else} \\ 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases} \quad (9)$$

From (9) we can see that TiMBL treats numeric and symbolic features differently. For numeric features the unweighted distance is computed as the difference between the values for that feature in each instance divided by the maximum possible distance computed over all pairs of instances in the database.⁵ For symbolic features the unweighted distance is 0 if they are identical and 1 otherwise. For both numeric and symbolic features this distance is multiplied by the weight for the particular feature, based on the Gain Ratio measure introduced by Quinlan (1993). This is a measure of the difference in uncertainty between the situations with and without knowledge of the value of that feature, as in (10).

$$w_i = \frac{H(C) - \sum_v \Pr(v) \times H(C|v)}{H(v)} \quad (10)$$

Where C is the set of classifications, v ranges over all values of the feature i and $H(C)$ is the entropy of the class labels. Probabilities are estimated from frequency of occurrence in the training data. The numerator of formula metric determines the knowledge about the distribution of classes which is added by knowing the value of feature i . However, this measure can overestimate the value of features with large numbers of possible values. To compensate it is divided by $H(v)$, the entropy of the feature-values.

Word senses are presented to TiMBL in a feature-vector representation with each sense which was not removed by the part of speech filter being represented by a separate vector. The vectors are formed from the following pieces of information in order: headword, homograph number, sense number, rank of sense (the order of the sense in the lexicon), part of speech from lexicon, output from the three partial taggers (**simulated annealing**, **subject codes** and **selectional restrictions**), surface form of headword from the text, the ten collocates and an indicator of whether the sense is appropriate or not in the context (**correct** or **incorrect**).

Table 5 shows a feature vectors generated for the word *influence* in the context shown. The final value in the feature vector shows whether the sense is correct or not in the particular context. We can see that, in this case, there is one correct sense, **influence_1_1a**, the definition of which is “power to gain an effect on the mind of or get results from, without asking or doing anything”. Features 10-19 are produced by the collocation extractor and these are identical since each vector is taken from the same content. Features 7-9 show the results of the partial taggers. The first is the output from **simulated annealing**, the second the **subject code** and the third the **selectional restrictions**. All noun senses of *influence* share the same pragmatic code (--) and consequently this partial tagger returns the same score for each sense. A final point worth noting is that in LDOCE *influence* has a verb sense, but as the part of speech filter removed this sense from consideration it is not included in the feature vector representation.

⁵ An earlier version of this system (Stevenson and Wilks, 1999) used TiMBL version 1.0 (Daelemans et al., 1998), which supports only symbolic features.

Context												
Regarding Atlanta's new million dollar airport, the jury recommended "that when the new management take charge Jan. 1 the airport be operated in a manner that will eliminate political influences ".												
Feature Vectors												
Learning features												Truth
influence 1 1a 1 n influences 1 12.03 y NoColl manner NoColl eliminate NoColl in NoColl political NoColl eliminate												correct
influence 1 1b 2 n influences 0 12.03 y NoColl manner NoColl eliminate NoColl in NoColl political NoColl eliminate												incorrect
influence 1 2 3 n influences 0 12.03 y NoColl manner NoColl eliminate NoColl in NoColl political NoColl eliminate												incorrect
influence 1 3 4 n influences 0 12.03 y NoColl manner NoColl eliminate NoColl in NoColl political NoColl eliminate												incorrect
influence 1 4 5 n influences 0 12.03 n NoColl manner NoColl eliminate NoColl in NoColl political NoColl eliminate												incorrect
influence 1 5 6 n influences 0 12.03 n NoColl manner NoColl eliminate NoColl in NoColl political NoColl eliminate												incorrect
influence 1 6 7 n influences 0 12.03 n NoColl manner NoColl eliminate NoColl in NoColl political NoColl eliminate												incorrect

Figure 5

Example feature vector representation

The TiMBL algorithm is trained on tokens presented in this format. When disambiguating unannotated text, the algorithm is applied to data presented in the same format without the classification. The unclassified vectors are then compared with all the training examples and it is then assigned the class of the closest one.

5 Evaluation Strategy

5.1 Evaluation Corpus

The evaluation of WSD algorithms has recently become a much-studied area. Gale, Church, and Yarowsky (1992a), Resnik and Yarowsky (1997) and Melamed and Resnik (2000) each presented arguments for adopting various evaluation strategies, with Resnik and Yarowsky's proposal directly influencing the set-up of SENSEVAL (Kilgarriff, 1998). At the heart of their proposals is the ability of human subjects to markup text with the phenomenon in question (WSD in this case) and evaluate the results of computation. This linguistic phenomenon has proved to be far more elusive and complex than many others. We have discussed this at length elsewhere (Wilks, 1997) and will assume here that humans can markup text for senses to a sufficient degree. Kilgarriff (1993) questioned the possibility of creating sense tagged texts, claiming the task to be impossible. However, it should be borne in mind that no alternative has yet been widely accepted and that Kilgarriff himself used the markup-and-test model for SENSEVAL. In the following discussion we compare the evaluation methodology adopted here with those proposed by others.

The standard evaluation procedure for WSD is to compare the output of the system against gold standard texts, but these are very labour-intensive to obtain; lexical semantic markup is generally accepted as a more difficult and time-consuming task than part-of-speech markup (Fellbaum et al., 1998). Rather than expend a vast amount of effort on manual tagging we decided to combine two existing resources. We took SEMCOR (Landes, Leacock, and Tengi, 1998), a 200,000 word corpus with the content words manually tagged as part of the WordNet project. The semantic tagging was carried out by trained lexicographers under disciplined conditions which attempted to keep tagging inconsistencies to a minimum. SENSUS (Knight and Luk, 1994) is a large-scale ontology designed for machine-translation and was produced by merging the ontological hierarchies of WordNet, LDOCE (as derived by Bruce and Guthrie, see Section 4.4) and the Penman Upper Model from ISI. To facilitate this merging of the three resources it was

necessary to derive a mapping between the senses in the two lexical resources. We used this mapping to translate the WordNet-tagged content words in SEMCOR to LDOCE tags.

The mapping of senses is not one-to-one, and some WordNet synsets are mapped onto two or three LDOCE senses when WordNet does not distinguish between them. The mapping also contained significant gaps, chiefly words and senses not in the translation scheme. SEMCOR contains 91,808 words tagged with WordNet synsets, 6,071 of which are proper names which we ignored, leaving 85,737 words which could potentially be translated. The translation contains only 36,869 words tagged with LDOCE senses, although this is a reasonable size for an evaluation corpus for the task; and it is several orders of magnitude larger than those used by other researchers working in large vocabulary WSD, for example Cowie, Guthrie, and Guthrie (1992), Harley and Glennon (1997) and Mahesh et al. (1997). This corpus was also constructed without the excessive cost of additional hand-tagging and does not introduce any of the inconsistencies that can occur with a poorly controlled tagging strategy.

Resnik and Yarowsky (1997) proposed to evaluate large vocabulary WSD systems by choosing a set of test words and providing annotated test and training examples for just these, allowing supervised and unsupervised algorithms to be tested on the same vocabulary. This model was implemented in SENSEVAL (Kilgarriff, 1998). However, for the evaluation of the system presented here there would have been no benefit from using this strategy since it still involves the manual tagging of large amounts of data and one can then evaluate over all words, as we do. It is possible that some computational techniques may evaluate well over a small vocabulary but may not work for a large set of words, and the evaluation strategy proposed by Resnik and Yarowsky will not discriminate between these cases.

Ambiguous tokens in our evaluation corpus occurred between 1 and 604 times (with “have” being the most frequent). A large number of words (2407) occur only once and nearly 95% have 25 occurrences or less. Table 6 shows the distribution of ambiguous types by number of corpus tokens. It is worth noting that, as would be expected, the observed distribution is highly Zipfian (Zipf, 1935).

Table 6
Occurrence of ambiguous words in evaluation corpus

Occurrence Range	Count
1-25	5488 (94.6%)
26-50	202 (3.5%)
51-75	67 (1.2%)
76-100	21 (0.04%)
100-604	26 (0.4%)

Differences in evaluation corpora makes comparison difficult. However, some idea of the difficulty of WSD can be gained by calculating properties of the evaluation corpus. Gale, Church, and Yarowsky (1992a) suggest that the lowest level of performance which can be reasonably expected from a WSD system is that achieved by assigning the most likely sense in all cases. Since the first sense in LDOCE is usually the most frequent, we use a heuristic which assumes the first sense is always correct to calculate this baseline figure. This is the same baseline heuristic we used for the experiments reported in Section 3, although those were for the homograph level. We applied the naive heuristic of always choosing the first sense in our corpus and found that 30.9% of senses were correctly

disambiguated.

Another measure which gives insight into an evaluation corpus is to count the average polysemy, the number of possible senses we can expect for each ambiguous word in the corpus. This is calculated by counting the sum of possible senses for each ambiguous token and dividing by the number of tokens. This is represented by (11) where w ranges over all ambiguous tokens in the corpus, $S(w)$ is the number of possible senses for word w and N is the number of ambiguous tokens. The average polysemy for our evaluation corpus is 14.62.

$$\text{Average polysemy} = \frac{\sum_{w \text{ in } text} S(w)}{N} \quad (11)$$

Our annotated corpus has the unusual property that more than one sense may be marked as correct for a particular token. This is an unavoidable side-effect of a mapping between lexicon senses which is not one-to-one. However, it does not imply that WSD is easier in this corpus than one in which only a single sense is marked for each token, as can be shown from an imaginary example. The worst case for a WSD algorithm is when each of the possible semantic tags for a given word occurs with equal frequency in a corpus (and so the prior probabilities exhibit a uniform, uninformative distribution). Then a corpus with an average polysemy of 5 and 2 senses marked correct on each ambiguous token will have a baseline not less than 40%. However, one with an average polysemy of 2 and only a single sense on each will have a baseline of at least 50%. Test corpora in which each ambiguous token has exactly two senses were used by Yarowsky (1995), Brown et al. (1991) and others.

Our system was tested using a technique known as 10-fold cross validation. This process is carried out by splitting the available data into ten roughly equal subsets. One of the subsets is chosen as the test data and the TiMBL algorithm is trained on the remainder. This is repeated ten times, so that each subset is used as test data exactly once, and results are averaged across all of the test runs. This technique provides two advantages: first, the best use can be made of the available data and, secondly, the computed results are more statistically reliable than those obtained by simply setting aside a single portion of the data for testing.

5.2 Evaluation Metrics

The choice of scoring metric is an important one in the evaluation of WSD algorithms. The most commonly used is the ratio of words for which the system has assigned the correct sense compared to those which it attempted to disambiguate. Resnik and Yarowsky (1997) dubbed this the exact match criterion which is usually expressed as a percentage calculated according to formula 12.

$$\text{Exact match} = \frac{\text{Number of correctly assigned senses}}{\text{Number of senses assigned}} \times 100\% \quad (12)$$

Resnik and Yarowsky criticise the exact match measure since it assumes a WSD system commits to a particular sense. They propose an alternative measure based on cross-entropy which compares the probabilities for each sense as assigned by a WSD system against those in the gold standard text. Formula 13 shows the method for computing this metric, where the WSD system has processed N words and $\text{Pr}(cs_i)$ is the probability assigned to the correct sense of word i .

$$-\frac{1}{N} \sum_{i=1}^N \log_2 \text{Pr}(cs_i) \quad (13)$$

This evaluation metric may be useful for disambiguation systems which assign probabilities to each sense, such as those developed by Resnik and Yarowsky, since it provides more information than the exact match metric. However, for systems which simply choose a single sense and do not measure confidence it provides far less information. When a WSD assigns only one sense to a word and that sense is incorrect, that word is scored as ∞ . Consequently, formula 13 returns ∞ if there is at least one word in the test set for which the tagger assigns a zero probability to the correct sense. For WSD systems which assign exactly one sense to each word this metric returns 0 if all words are tagged correctly and ∞ otherwise. This metric is potentially very useful for the evaluation of WSD systems which return (non-zero) probabilities for each possible sense but not for the one presented in this paper and others which are not based on probabilistic models.

Melamed and Resnik (2000) propose a metric for scoring WSD output when there may be more than one correct sense in the gold standard text, as with the evaluation corpus we use. They mention that when a WSD system returns more than one sense it is difficult to tell if they are intended to be disjunctive or conjunctive. The score for a token is computed by dividing the number of correct senses identified by the algorithm by the total it returns and so the metric is equivalent to precision in information retrieval (van Rijsbergen, 1979).⁶ For systems which return exactly one sense for each word this equates to scoring a token as 1 if the sense returned is correct and 0 otherwise. For the evaluation of the system presented here the metric proposed by Melamed and Resnik is then equivalent to the exact match metric.

The exact match metric has the advantage of being widely used in the WSD literature. In our experiments the exact match figure is computed at the LDOCE sense level, where the number of tokens correctly disambiguated to the sense level is divided by the number ambiguous at that level. At the homograph level, the number correctly disambiguated to the homograph is divided by the number which are polyhomographic.

6 Performance

Using the evaluation procedure described in the previous section it was found that the system correctly disambiguated 90% of the ambiguous instances to the fine-grained sense level and in excess of 94% to the homograph level.

In order to analyse the effectiveness of our tagger in more detail, we split the main corpus into sub-corpora by grammatical category. In other words, we created four individual sub-corpora containing the ambiguous words which had been part-of-speech tagged as nouns, verbs, adjectives and adverbs. The figures characterising each of these corpora are shown in Table 7. The majority of the ambiguous words were nouns with far fewer verbs and adjectives and less than one thousand adverbs. The average polysemy for nouns, at both sense and homograph levels, is roughly the same as the overall corpus average although it is noticeably higher for verbs at the sense level. At the sense level the average polysemy figures are much lower for adjectives and adverbs. This is because it is common for English words to act as either a noun or a verb and, since these are the most polysemous grammatical categories, the average polysemy count becomes large due to the cumulative effect of polysemy across grammatical categories. However, words which can act as adjectives or adverbs are unlikely to be nouns or verbs. This, plus the fact that adjectives and adverbs are generally less polysemous in LDOCE, means that their average polysemy in text is far lower than it is for nouns or verbs.

Table 7 shows the accuracy of our system over the four sub-corpora. We can see that

⁶ The metric operates slightly differently for systems which assign probabilities to senses.

Table 7

System results, baselines and corpus characteristics. Sense level results are calculated over all polysemous words in the evaluation corpus while those reported for the homograph level only over polyhomographic ones.

		Entire Corpus	Noun	Sub-corpora		
				Verb	Adjective	Adverb
Sense level	Accuracy	90.37%	91.24%	88.38%	91.09%	70.61%
	Baseline	30.90%	34.56%	18.46%	25.76%	36.73%
Tokens		36,774	26,091	6,465	3,310	908
Types		5,804	4,041	1,021	1,006	125
Average Polysemy		14.62	13.65	24.35	6.07	4.43
Homograph level	Accuracy	94.65%	94.63%	95.26%	96.89%	90.67%
	Baseline	71.24%	73.47%	60.72%	87.10%	86.87%
Tokens		18,219	11,380	5,194	1,326	319
Types		1,683	1,264	709	201	34
Average Polysemy		2.52	2.32	2.81	2.95	3.13

the tagger achieves higher results at the homograph level than the sense level on each of the four sub-corpora which is consistent with the result over the whole corpus.

There is quite a difference in the tagger’s results across the different sub-corpora; 91% for nouns and 70% for adverbs, perhaps because the learning algorithm does not perform as well on adverbs since that corpus is significantly smaller than the other three. This hypothesis was checked by testing our system on portions of each of the three sub-corpora which were of roughly equal size to the adverb sub-corpus. We found that the reduced data caused a slight loss of accuracy on each of the three sub-corpora, however there was still a marked difference between the results for the adverb sub-corpus and the other three. Further analysis showed that the differences in performance over different sub-corpora seem linked to the behaviour of different partial taggers in combination which we now go on to describe in more detail.

6.1 Interaction of Knowledge Sources

In order to gauge the contribution of each knowledge source separately, we implemented a set of simple disambiguation algorithms each of which uses the output from a single partial tagger. Each algorithm takes the result of its partial tagger and checks against the disambiguated text to see if it is correct. If the partial tagger returns more than one sense, such as the `simulated annealing`, `subject code` and `selectional preference` taggers, the first sense is taken to break the tie. For the partial tagger based on Yarowsky’s subject code algorithm we choose the sense with the highest saliency value and, again, if more than one sense has been assigned the maximum value, the first is chosen. Therefore, each partial tagger returns a single sense and the exact match measure is used to determine how many tokens they correctly disambiguate. The part-of-speech filter is run before the partial taggers make their decision and so they only consider the set of senses it did not remove. The results, at both sense and homograph level, of each tagger computed over the evaluation corpus and four sub-corpora are shown in Table 7.

We can see that the partial taggers which are most effective are those based on the simulated annealing algorithm and Yarowsky’s subject code approach. The success of

Table 8
Performance of individual partial taggers (at sense level)

	All	Nouns	Verbs	Adjectives	Adverbs
simulated annealing (1)	65.24%	66.50%	67.51%	49.02%	50.61%
selectional preferences (2)	44.85%	40.73%	75.80%	27.56%	0%
subject codes (3)	79.41%	79.18%	72.75%	73.73%	85.50%

these modules supports our decision to use existing disambiguation algorithms which have already been developed rather than creating new ones.

The most successful of the partial taggers is the one based on Yarowsky’s algorithm for modelling thesaural categories by wide contexts. This consistently achieves over 70% correct disambiguation and seems particularly successful when disambiguating adverbs (over 85% correct). It is quite surprising that this algorithm is so successful for adverbs since it could seem quite reasonable to expect an algorithm based on subject codes to be more successful on nouns and less so on modifiers such as adjectives and adverbs.

Yarowsky (1992) reports that his algorithm achieves 92% correct disambiguation which is nearly 13% higher than achieved in our implementation. However, Yarowsky tested his implementation on just 12 words, the majority of which were nouns, and used Roget large categories as senses. Since thesauri tend to contain more coarse-grained semantic distinctions than dictionaries, there is a lower average polysemy of 3.00 categories per word. The baseline performance for this corpus is 66.5%, considerably higher than the 30.9% computed for the corpus used in our experiments. Another possible reason for the difference in results is the fact that Yarowsky used smoothing algorithms to avoid problems with the probability estimates caused by data sparseness. We did not employ these procedures and used simple corpus frequency counts when calculating the probabilities (see Section 4.5). It is not possible to say that the differences between implementations did not lead to the differences in results but it seems likely that the difference in the semantic granularity of LDOCE subject codes and Roget categories was an important factor.

The second partial tagger based on an existing approach is the one which uses the simulated annealing algorithm. In Section 4.3 we noted that Cowie et. al. reported 47% correct disambiguation to the sense level for this algorithm. However, in our application of the algorithm over 17% more words are correctly disambiguated. But the original application of the algorithm did not filter out senses with the incorrect part of speech and we changed the method by which overlap is calculated to take account of short definitions and these changes are likely to be the source of the improved results.

Our least successful partial tagger is the one based on selectional preferences. Although its overall result is slightly below the overall corpus baseline, it is very successful when disambiguating verbs. This is consistent with the work of Resnik (1997) who reported that many words do not have strong enough selectional restrictions to carry out WSD. We expected preferences to be successful for adjectives as well although this is not the case in our evaluation. This is because the sense discrimination of adjectives is carried out after that for nouns in our algorithm (see Section 4.4) and the first is hindered by the low results of the second. Adverbs cannot be disambiguated by preference methods against LDOCE since it does not contain the appropriate information.

Our analysis of the behaviour of the individual partial taggers provides some clues to the behaviour of the tagger on the different sub-corpora. In general, the tagger performs to a roughly similar level on the noun, verb and adjective sub-corpus, there is less than 3% difference between the best and worse performance. The taggers' worst performance is on the adverb sub-corpus where it disambiguates less than 70% of the tokens successfully. This seems to be due to the poor performance of the individual partial taggers on that grammatical category. However, the tagger still manages to disambiguate most of the adverbs which are ambiguous at the homograph level successfully and this is probably because the part-of-speech filter has ruled out the incorrect homographs rather than the performance of the partial tagger.

One can legitimately wonder whether in fact the different knowledge sources for WSD are all ways of encoding the same semantic information, as one can believe that part-of-speech tagging rules and stochastic methods encode the same information about tag sequences in different formats. However, the fact that an optimised combination of our partial taggers yield a significantly higher figure than any operating independently shows that they must be orthogonal information sources.

6.2 The overall value of the Part-of-speech filter

We have already examined the usefulness of part-of-speech tags for semantic disambiguation in Section 3. However, we now want to know the effect it has within a system consisting of several disambiguation modules. It was found that accuracy at the sense level reduced to 87.87% and to 93.36% at the homograph level when the filter was removed. Although the system's performance did not decrease by a large amount, the part-of-speech filter brings the additional benefit of reducing the search space for the three partial taggers. In addition, the fact that these results are not much effected by the removal of the part-of-speech filter shows that the WSD modules alone do a reasonable job of resolving part-of-speech ambiguity as a side-effect of semantic disambiguation.

7 Conclusion

Previously reported WSD systems which enjoyed a high level of accuracy have operated on a very restricted vocabulary and often with a single WSD methodology. These methods have often been pursued for sound reasons to do with evaluation, but have limited their applicability (as part of, say, an information extraction system) and also their persuasiveness regarding the scalability and interaction of the various WSD partial methods. This paper reported a system which disambiguated all content words in a text, as defined by a standard machine readable dictionary, with a high degree of accuracy.

Our evaluation shows that disambiguation can be carried out with more accurate results when several knowledge sources are combined. It remains unclear exactly what it means to optimise the combination of modules within a learning system like `TiMBL`: we could, in further work, treat the part-of-speech tagger as a partial tagger and not a filter, and we could allow the system to learn some "optimal" weighting of all the partial taggers. It also remains an interesting questions, whether, because of the undoubted existence of novel senses in text, a sense tagger can ever reach the levels that part-of-speech tagging has. However, we believe we have shown that interesting combinations of WSD methods on a substantial training corpus are possible, and that this can show, among other things, the relative independence of the types of semantic information expressed by the various forms of lexical input.

Acknowledgments

The work described here was supported by the European Union Language Engineering project ECRAN – Extraction of Content: Research at Near-market (LE-2110). One of the authors was also supported by the EPSRC grant MALT (GR/M73521) while writing this paper. We are grateful for the feedback from many colleagues in Sheffield especially Mark Hepple and the detailed comments from the anonymous reviewers of an earlier version of this paper. Any errors are our own.

References

- Brill, E. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–566.
- Brown, P., S. Della Pietra, V. Della Pietra, and R. Mercer. 1991. Word sense disambiguation using statistical methods. In *Proceedings of the 29th Meeting of the Association for Computational Linguistics (ACL-91)*, pages 264–270, Berkeley, C.A.
- Bruce, R. and L. Guthrie. 1992. Genus disambiguation: A study in weighted preference. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 1187–1191, Nantes, France.
- Bruce, R. and J. Wiebe. 1994. Word-sense disambiguation using decomposable models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL-94)*, pages 139–145, Las Cruces, New Mexico.
- Burnard, L., 1995. *Users Reference Guide for the British National Corpus*. Oxford University Computing Services.
- Cost, S. and S. Salzberg. 1993. A weighted nearest neighbour algorithm for learning with symbolic features. *Machine Learning*, 10(1):57–78.
- Cottrell, G. 1984. A model of lexical access of ambiguous words. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-84)*, pages 61–67.
- Cowie, J., L. Guthrie, and J. Guthrie. 1992. Lexical disambiguation using simulated annealing. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 359–365, Nantes, France.
- Daelemans, W., J. Zavrel, P. Berck, and S. Gillis. 1996. MBT: A memory-based part of speech tagger generator. In *Proceedings of the Fourth Workshop on Very Large Corpora*, pages 14–27.
- Daelemans, W., J. Zavrel, K. van der Sloot, and A. van den Bosch. 1998. TiMBL: Tilburg memory based learner version 1.0. Technical report, ILK Technical Report 98-03.
- Daelemans, W., J. Zavrel, K. van der Sloot, and A. van den Bosch. 1999. TiMBL: Tilburg memory based learner, version 2.0, reference guide. Technical report, ILK Technical Report 99-01. Available from <http://ilk.kub.nl/~ilk/papers/ilk9901.ps>.
- Fellbaum, C., J. Grabowski, S. Landes, and A. Baumann. 1998. Matching words to senses in WordNet: Naive vs. expert differentiation of senses. In C. Fellbaum, editor, *WordNet: An electronic lexical database and some applications*. MIT Press, Cambridge, MA.
- Gaizauskas, R., T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. 1996. Description of the LaSIE system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 207–220, San Francisco, CA.
- Gale, W., K. Church, and D. Yarowsky. 1992a. Estimating upper and lower bounds on the performance of word sense disambiguation programs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL-92)*, pages 249–256, Newark, DE.
- Gale, W., K. Church, and D. Yarowsky. 1992b. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439.
- Gale, W., K. Church, and D. Yarowsky. 1992c. One sense per discourse. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 233–237, Harriman, NY.
- Guo, C.-M. 1989. Constructing a Machine Tractable Dictionary from Longman Dictionary of Contemporary English. Technical Report MCCS-89-156, Computing Research Laboratory, New Mexico State University.
- Harley, A. and D. Glennon. 1997. Sense tagging in action: Combining different tests with additive weights. In *Proceedings of the SIGLEX Workshop “Tagging Text with Lexical Semantics”*, pages 74–78, Washington, DC.
- Hirst, G. 1987. *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge, UK.

- Hirst, G. 1995. Near-synonymy and the structure of lexical knowledge. In *American Association for Artificial Intelligence Spring Symposium on Lexicons*, pages 51–56.
- Ide, N. and J. Véronis. 1998. Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1–40.
- Kilgarrieff, A. 1993. Dictionary word sense distinctions: An enquiry into their nature. *Computers and the Humanities*, 26:356–387.
- Kilgarrieff, A. 1997. Sample the lexicon. Technical Report Report ITRI-97-01, ITRI, University of Brighton.
- Kilgarrieff, A. 1998. SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs. In *Proceedings of the First International conference on Language Resources and Evaluation*, pages 581–585, Granada, Spain.
- Knight, K. and S. Luk. 1994. Building a large knowledge base for machine translation. In *Proceedings of the American Association for Artificial Intelligence Conference (AAAI-94)*, pages 185–199, Seattle, WA.
- Kučera, H. and W. Francis. 1967. *A Computational Analysis of Present-day American English*. Brown University Press, Providence, RI.
- Landes, S., C. Leacock, and R. Teng. 1998. Building a semantic concordance of English. In C. Fellbaum, editor, *WordNet: An electronic lexical database and some applications*. MIT Press, Cambridge, MA.
- Leacock, C., G. Towell, and E. Voorhees. 1993. Corpus-based statistical sense resolution. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 260–265, Plainsboro, NJ.
- Lesk, M. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of ACM SIGDOC Conference*, pages 24–26, Toronto, Canada.
- Mahesh, K., S. Nirenburg, S. Beale, E. Viegas, V. Raskin, and B. Onyshkevych. 1997. Word sense disambiguation: Why have statistics when we have these numbers? In *Proceedings of the 7th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 151–159, Santa Fe, NM.
- Marcus, M., B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Tree Bank. *Computational Linguistics*, 19(2):313–330.
- Masterman, M. 1957. The thesaurus in syntax and semantics. *Mechanical Translation*, 4:1–2.
- McRoy, S. 1992. Using multiple knowledge sources for word sense disambiguation. *Computational Linguistics*, 18(1):1–30.
- Melamed, I. and P. Resnik. 2000. Evaluation of sense disambiguation given hierarchical tag sets. *Computers and the Humanities*, 34:1–2.
- Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. 1953. Equation state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.
- Nadas, A., D. Nahamoo, M. Picheny, and J. Powell. 1991. An iterative “flip-flop” approximation of the most informative split in the construction of decision trees. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 565–568, Toronto, Canada.
- Ng, H. and H. Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Meeting of the Association for Computational Linguistics (ACL-96)*, pages 40–47, Santa Cruz, CA.
- Press, W., S. Teukolsky, W. Vetterling, and B. Flannery. 1988. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge.
- Procter, P., editor. 1978. *Longman Dictionary of Contemporary English*. Longman Group, Essex, UK.
- Procter, P., editor. 1995. *Cambridge International Dictionary of English*. Cambridge University Press, Cambridge.
- Quinlan, J. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.
- Resnik, P. 1997. Selectional preferences and word sense disambiguation. In *Proceedings of the SIGLEX Workshop “Tagging Text with Lexical Semantics: What, why and how?”*, pages 52–57, Washington, D.C.
- Resnik, P. and D. Yarowsky. 1997. A perspective on word sense disambiguation techniques and their evaluation. In *Proceedings of the SIGLEX Workshop “Tagging Text with Lexical Semantics: What, why and how?”*, pages 79–86, Washington, D.C.

- Rigau, G., J. Atserias, and E. Agirre. 1997. Combining unsupervised lexical knowledge methods for word sense disambiguation. In *35th Meeting of the Association for Computational Linguistics and the 8th Meeting of the European Chapter of the Association for Computational Linguistics (ACL/EACL-97)*, pages 48–55, Madrid, Spain.
- Schütze, H. 1992. Dimensions of meaning. In *Proceedings of Supercomputing '92*, pages 787–796, Minneapolis, MN.
- Stevenson, M. 1998. Extracting syntactic relations using heuristics. In *Proceedings of the European Summer School on Logic, Language and Information '98 Student Workshop*, pages 248–256, Saarbrücken, Germany.
- Stevenson, M. and Y. Wilks. 1999. Combining weak knowledge sources for sense disambiguation. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 884–889, Stockholm, Sweden.
- van Rijsbergen, C. 1979. *Information Retrieval*. Butterworths, London.
- Véronis, J. and N. Ide. 1990. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, pages 389–394, Helsinki, Finland.
- Waltz, D. and J. Pollack. 1985. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, 9:51–74.
- Wilks, Y. 1975. A preferential pattern-seeking semantics for natural language inference. *Artificial Intelligence*, 6:53–74.
- Wilks, Y. 1997. Senses and Texts. *Computers and the Humanities*, 31:77–90.
- Wilks, Y., D. Fass, C.-M. Guo, J. McDonald, T. Plate, and B. Slator. 1990. Providing machine tractable dictionary tools. *Machine Translation*, 5:99–154.
- Wilks, Y., B. Slator, and L. Guthrie. 1996. *Electric Words: Dictionaries, Computers and Meanings*. MIT Press, Cambridge, MA.
- Wilks, Y. and M. Stevenson. 1998a. The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Journal of Natural Language Engineering*, 4(2):135–144.
- Wilks, Y. and M. Stevenson. 1998b. Optimising combinations of knowledge sources for word sense disambiguation. In *Proceedings of the 36th Meeting of the Association for Computational Linguistics (COLING-ACL-98)*, pages 1398–1402, Montreal, Canada.
- Yarowsky, D. 1992. Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 454–460, Nantes, France.
- Yarowsky, D. 1993. One sense per collocation. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 266–271, Princeton, NJ.
- Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, pages 189–196, Cambridge, MA.
- Yarowsky, D. 1996. Homograph disambiguation in text-to-speech synthesis. In J. Hirschberg, R. Sproat, and J. van Santen, editors, *Progress in Speech Synthesis*. Springer-Verlag, pages 159–175.
- Yngve, V. 1955. Syntax and the problem of multiple meaning. In W. Locke and D. Booth, editors, *Machine Translation of Languages*. Wiley, New York.
- Zavrel, J., W. Daelemans, and J. Veenstra. 1997. Resolving PP-attachment with memory-based learning. In *Proceedings of the Workshop on Computational Natural Language Learning (CoNLL '97)*, pages 136–144, Madrid, Spain.
- Zipf, G. 1935. *The Psycho-Biology of Language*. Houghton Mifflin, Boston.