17 WORD SENSE DISAMBIGUATION AND INFORMATION RETRIEVAL

Oh are you from Wales?
Do you know a fella named Jonah?
He used to live in whales for a while.
Groucho Marx

This chapter introduces a number of topics related to **lexical semantic processing**. By this, we have in mind applications that make use of word meanings, but which are to varying degrees decoupled from the more complex tasks of compositional sentence analysis and discourse understanding.

LEXICAL SEMANTIC PROCESSING

The first topic we cover, **word sense disambiguation**, is of considerable theoretical and practical interest. As we noted in Chapter 16, the task of word sense disambiguation is to examine word tokens in context and specify which sense of each word is being used. As we will see in the next two sections, making this vague definition operational is a non-trivial — there is no clear consensus as to exactly what the task is, or how it should be evaluated. Nevertheless, there are robust algorithms that can achieve high levels of accuracy under certain reasonable assumptions.

WORD SENSE DISAMBIGUATION

The second topic we cover, **information retrieval**, is an extremely broad field, encompassing a wide-range of topics pertaining to the storage, analysis, and retrieval of all manner of media (Baeza-Yates and Ribeiro-Neto, 1999). Our concern in this chapter is solely with the storage and retrieval of text documents in response to users requests for information. We are interested in approaches in which users' needs are expressed as words, and documents are represented in terms of the words they contain. Section 17.3 presents the **vector space model**, a well-established approach used in most current systems, including most Web search engines.

INFORMATION RETRIEVAL

17.1 Selection Restriction-Based Disambiguation

For the most part, our discussions of compositional semantic analyzers in Chapter 15 ignored the issue of lexical ambiguity. By now it should be clear that this is not a reasonable approach. Without some means of selecting correct senses for the words in the input, the enormous amount of homonymy and polysemy in the lexicon will quickly overwhelm any approach in an avalanche of competing interpretations. As with syntactic part-of-speech tagging, there are two fundamental approaches to handling this ambiguity problem. In the first approach, the selection of correct senses occurs during semantic analysis as a side-effect of the elimination of ill-formed representations composed from an incorrect combination of senses. In the second approach, sense disambiguation is performed as a stand-alone task independent of, and prior to, compositional semantic analysis. This section discusses the role of selection restrictions in the former approach. The stand-alone approach is discussed in detail in 17.2.

Selection restrictions and type hierarchies are the primary knowledge-sources used to perform disambiguation in most integrated approaches. In particular, they are used to rule out inappropriate senses and thereby reduce the amount of ambiguity present during semantic analysis. If we assume an integrated rule-to-rule approach to semantic analysis, then selection restrictions can be used to block the formation of component meaning representations that contain violations. By blocking such ill-formed components, the semantic analyzer will find itself dealing with fewer ambiguous meaning representations. This ability to focus on correct senses by eliminating flawed representations that result from incorrect senses can be viewed as a form of indirect word sense disambiguation. While the linguistic basis for this approach can be traced back to the work of Katz and Fodor (1963), the most sophisticated computational exploration of it is due to Hirst (1987).

As an example of this approach, consider the following pair of WSJ examples, focusing solely on their use of the lexeme *dish*.

(17.1) "In our house, everybody has a career and none of them includes washing dishes", he says.

(17.2) In her tiny kitchen at home, Ms. Chen works efficiently, stir-frying several simple dishes, including braised pig's ears and chicken livers with green peppers.

These examples make use of two polysemous senses of the lexeme *dish*. The first refers to the physical objects that we eat from, while the second refers to

the actual meals or recipes. The fact that we perceive no ambiguity in these examples can be attributed to the selection restrictions imposed by *wash* and *stir-fry* on their PATIENT roles, along with the semantic type information associated with the two senses of *dish*. More specifically, the restrictions imposed by *wash* conflict with the food sense of dish since it does not denote something that is normally washable. Similarly, the restrictions on *stir-fry* conflict with the artifact sense of dish, since it does not denote something edible. Therefore, in both of these cases *the predicate selects the correct sense* of an ambiguous argument by eliminating the sense that fails to match one of its selection restrictions.

Now consider the following WSJ and ATIS examples, focusing on the ambiguous predicate *serve*.

(17.3) Well, there was the time they served green-lipped mussels from New Zealand.

(17.4) Which airlines serve Denver?

(17.5) Which ones serve breakfast?

Here the sense of *serve* in 17.3 requires some kind of food as its PATIENT, the sense in 17.4 requires some kind of geographical or political entity, and the sense in the last example requires a meal designator. If we assume that *mussels*, *Denver* and *breakfast* are unambiguous, then in it is the arguments in these examples that select the appropriate sense of the verb.

Of course, there are also cases where both the predicate and the argument have multiple senses. Consider the following BERP example.

(17.6) I'm looking for a restaurant that serves vegetarian dishes.

Restricting ourselves to three senses of *serve* and two senses of *dish* yields six possible sense combinations in this example. However, since only one combination of the six is free from a selection restriction violation, determining the correct sense of both *serve* and *dish* is straightforward. In particular, the predicate and argument mutually select the correct senses.

Before moving on, we should note there will always be examples like the following where the available selection restrictions are too general to uniquely select a correct sense.

(17.7) What kind of dishes do you recommend?

In cases like this we either have to rely on the stand-alone methods discussed in 17.2, or knowledge of the broader discourse context, as will be discussed in Chapter 18.

Although there are a wide variety of ways to integrate this style of disambiguation into a semantic analyzer, the most straightforward approach follows the rule-to-rule strategy introduced in Chapter 15. In this integrated approach, fragments of meaning representations are composed and checked for selection restriction violations as soon as their corresponding syntactic constituents are created. Those representations that contain selection restriction violations are eliminated from further consideration.

This approach requires two additions to the knowledge structures used in our semantic analyzers: access to hierarchical type information about the arguments, and semantic selection restriction information about the arguments to predicates . Recall from Chapter 16, that both of these can be encoded using knowledge from WordNet. The first is available in form of the hypernym information about the heads of the meaning structures being used as arguments to predicates. Similarly, selection restriction information about argument roles can be encoded by associating the appropriate WordNet synsets with the arguments to each predicate-bearing lexical item. Exercise **??** asks you to explore this approach in more detail.

Limitations of Selection Restrictions

Not surprisingly, there are a number of practical and theoretical problems with this use of selection restrictions. The first symptom of these problems is the fact that there are many perfectly well-formed, interpretable, sentences that contain obvious violations of selection restrictions. Therefore, any approach based on a strict *elimination* of such interpretations is in serious trouble.

Consider the following WSJ example.

(17.8) But it fell apart in 1931, perhaps because people realized you can't eat gold for lunch if you're hungry.

The phrase *eat gold* clearly violates the selection restriction that *eat* places on its PATIENT role. Nevertheless, this example is perfectly well-formed. The key is the negative environment set up by *can't* prior to the violation of the restriction. This example makes it clear that any purely local, or rule-to-rule, analysis of selection restrictions will fail when a wider context makes the violation of a selection restriction acceptable, as in this case.

A second problem with selection restrictions is illustrated by the following example.

(17.9) In his two championship trials, Mr. Kulkarni ate glass on an empty stomach, accompanied only by water and tea.

Although the event described in this example is somewhat unusual, the sentence itself is not semantically ill-formed, despite the violation of *eat*'s selection restriction. Examples such as this illustrate the fact that thematic roles and selection restrictions are merely loose approximations of the deeper concepts they represent. They can not hope to account for uses such as this that require deeper commonsense knowledge about what eating is all about. At best, they reflect the idea that the things that are eaten are normally edible.

Finally, as discussed in Chapter 16, metaphoric and metonymic uses challenge this approach as well. Consider the following WSJ example.

(17.10) If you want to kill the Soviet Union, get it to try to eat Afghanistan.

Here the typical selection restrictions on the PATIENTS of both *kill* and *eat* will eliminate all possible literal senses leaving the system with no possible meanings. In many systems, such a situation serves to trigger alternative mechanisms for interpreting metaphor and metonymy (Fass, 1997).

As Hirst (1987) observes, examples like these often result in the elimination of all senses, bring semantic analysis to a halt. One approach to alleviating this problem is to adopt the view of selection restrictions as preferences, rather than rigid requirements. Although there have been many instantiations of this approach over the years (Wilks, 1975c, 1975b, 1978), the one that has received the most thorough empirical evaluation is Resnik's (1998) work, which uses the notion of a *selectional association* introduced on page **??**. Recall that this notion uses an empirically derived measure of the strength of association between a predicate and a class dominating the argument to the predicate.

A simplified version of Resnik's disambiguation algorithm is shown in Figure 17.1. The basic notion behind this algorithm is to select as the correct sense for the argument, the one that has the highest selectional association between one of its ancestor hypernyms and the predicate. Resnik (1998) reports an average of 44% correct with this technique for verb-object relationships, a result that is an improvement over a most frequent sense baseline. A limitation of this approach is that it only addresses the case where the predicate is unambiguous and *selects* the correct sense of the argument. A more complex decision criteria would be needed for the more likely situation where both the predicate and argument are ambiguous.

```
function SA-WSD(pred, arg) returns sense

 best-association ← Mininum possible  selection association
 for each sense in senses of arg do
   for each hypernym in hypernyms of sense do
     new ← Selectional association between hyp and pred
     if new > best-association then
       best-association ← new
       best-sense ← sense
   end
 end
 return best-sense
```

Figure 17.1 Resnik's (1998) selectional association-based word sense disambiguation algorithm. The selection association between all the hypernyms of all the senses of the target argument and the predicate are computed. The sense with the most closely associated hypernym is selected.

17.2 Robust Word Sense Disambiguation

The selection restriction approach to disambiguation has too many requirements to be useful in large-scale practical applications. Even with the use of WordNet, the requirements of complete selection restriction information for all predicate roles, and complete type information for the senses of all possible fillers are unlikely to be met. In addition, as we saw in Chapters 10, 12, and 15, the availability of a complete and accurate parse for all inputs is unlikely to be met in environments involving unrestricted text.

To address these concerns, a number of robust disambiguation systems with more modest requirements have been developed over the years. As with part-of-speech taggers, these systems are designed to operate in a stand-alone fashion and make minimal assumptions about what information will be available from other processes.

Machine Learning Approaches

In machine learning approaches, systems are *trained* to perform the task of word sense disambiguation. In these approaches, what is learned is a classifier that can be used to assign as yet unseen examples to one of a fixed number of senses. As we will see, these approaches vary as to the nature

of the training material, how much material is needed, the degree of human intervention, the kind of linguistic knowledge used, and the output produced. What they all share is an emphasis on acquiring the knowledge needed for the task from data, rather than from human analysts. The principal question to keep in mind as we explore these systems is whether the method scales; that is, would it be possible to apply the method to a substantial part of the entire vocabulary of a language?

The Inputs: Feature Vectors

Before discussing the algorithms, we should first characterize the kind of inputs they expect. In most of these approaches, the initial input consists of the word to be disambiguated, which we will refer to as the **target** word, along with a portion of the text in which it is embedded, which we will call its **context**. This initial input is then processed in the following ways:

- The input is normally part-of-speech tagged using one of the high accuracy methods described in Chapter 8.
- The original context may be replaced with larger or smaller segments surrounding the target word.
- Often some amount of stemming, or more sophisticated morphological processing, is performed.
- Less often, some form of partial parsing, or dependency parsing, is performed to ascertain thematic or grammatical roles and relations.

After this initial processing, the input is then boiled down to a fixed set of features that capture information relevant to the learning task. This task consists of two steps: selecting the relevant linguistic features, and encoding them in a form usable in a learning algorithm. Fortunately, a simple **feature vector** consisting of numeric or nominal values can easily encode the most frequently used linguistic information, and is appropriate for use in most learning algorithms

FEATURE VECTOR

The linguistic features used in training WSD systems can be roughly divided into two classes: collocational features and co-occurrence features. In general, the term **collocation** refers to a quantifiable position-specific relationship between two lexical items. Collocational features encode information about the lexical inhabitants of *specific* positions located to the left and right of the target word. Typical items in this category include the word, the root form of the word, and the word's part-of-speech. This type of feature is effective at encoding local lexical and grammatical information that can often accurately isolate a given sense.

COLLOCATION

As an example of this type of feature-encoding, consider the situation where we need to disambiguate the lexeme *bass* in the following example.

(17.11) An electric guitar and **bass** player stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.

A feature-vector consisting of the two words to the right and left of the target word, along with their respective parts-of-speech, would yield the following vector.

```
[guitar, NN1, and, CJC, player, NN1, stand, VVB]
```

The second type of feature consists of co-occurrence data about neighboring words, ignoring their exact position. In this approach, the words themselves (or their roots) serve as features. The value of the feature is the number of times the word occurs in a region surrounding the target word. This region is most often defined as a fixed size window with the target word at the center. To make this approach manageable, a small number of frequently used content words are selected for use as features. This kind of feature is effective at capturing the general topic of the discourse in which the target word has occurred. This, in turn, tends to identify senses of a word that are specific to certain domains.

For example, a co-occurrence vector consisting of the 12 most frequent content words from a collection of *bass* sentences drawn from the WSJ corpus would have the words as features: *fishing, big, sound, player, fly, rod, pound, double, runs, playing, guitar, band*. Using these words as features with a window size of 10, Example 17.11 would be represented by the following vector.

```
[0,0,0,1,0,0,0,0,0,0,1,0]
```

As we will see, most robust approaches to sense disambiguation make use of a combination of both collocational and co-occurrence features.

Supervised Learning Approaches

In supervised approaches, a sense disambiguation system is learned from a representative set of labeled instances drawn from the same distribution as the test set to be used. This is a straightforward application of the **supervised learning** approach to creating a classifier. In such approaches, a learning system is presented with a training set consisting of feature-encoded inputs *along with their appropriate label, or category*. The output of the system is a classifier system capable of assigning labels to new feature-encoded inputs.

SUPERVISED LEARNING

METHODOLOGY BOX: EVALUATING WSD SYSTEMS

The basic metric used in evaluating sense disambiguation systems is simple precision: the percentage of words that are tagged correctly. The primary baseline against which this metric is compared is the **most frequent sense** metric: how well would a system do if it simply chose the most frequent sense of a word.

The use of precision requires access to the correct answers to the words in a test set. Fortunately, two large sense-tagged corpora are now available: the SEMCOR corpus (Landes *et al.*, 1998), which consists of a portion of the Brown corpus tagged with WordNet senses, and the SENSEVAL corpus (Kilgarriff and Rosenzweig, 2000), which is a tagged corpus derived from the HECTOR corpus and dictionary project.

A number of issues must be taken into account in comparing results across systems. The main issue concerns the nature of the senses used in the evaluation. Two approaches have been followed over the years: coarse distinctions among homographs, such as the musical and fish senses of *bass*, and fine-grained sense distinctions such as those found in traditional dictionaries. Unfortunately, there is no standard way of comparing results across these two kinds of efforts, or across efforts using different dictionaries.

Dictionary senses provide the opportunity for a more fine-grained scoring metric than simple precision. For example, confusing a particular musical sense of *bass* with a fish sense, is clearly worse than confusing it with another musical sense. This observation gives rise to a notion of **partial credit** in evaluating these systems. With such a metric, an exact sense-match would receive full credit, while selecting a broader sense would receive partial credit. Of course, this kind of scheme is entirely dependent on the organization of senses in the particular dictionary being used.

Standardized evaluation frameworks for word sense disambiguation systems are now available. In particular, the SENSEVAL effort (Kilgarriff and Palmer, 2000), provides the same kind of evaluation framework for sense disambiguation, that the MUC (Sundheim, 1995b) and TREC (Voorhees and Harman, 1998) evaluations have provided for information extraction and information retrieval.

Bayesian classifiers (Duda and Hart, 1973), decision lists (Rivest, 1987), decision trees (Quinlan, 1986), neural networks (Rumelhart *et al.*, 1986), logic learning systems (Mooney, 1995), and nearest neighbor methods(Cover and Hart, 1967) all fit into this paradigm. We will restrict our discussion to the naive Bayes and decision list approaches, since they have been the focus of considerable work in word sense disambiguation.

NAIVE BAYES

The **naive Bayes** classifier approach to WSD is based on the premise that choosing the best sense for an input vector amounts to choosing the most probable sense given that vector. In other words:

$$\hat{s} = \underset{s \in S}{\text{argmax}} P(s|V) \tag{17.12}$$

In this formula, S denotes the set of senses appropriate for the target associated with this vector. As is almost always the case, it would be difficult to collect statistics for this equation directly. Instead, we rewrite it in the usual Bayesian manner as follows:

$$\hat{s} = \underset{s \in S}{\text{argmax}} \frac{P(V|s)P(s)}{P(V)} \tag{17.13}$$

Of course, the data available that associates specific vectors with senses is too sparse to be useful. What is provided in abundance in the training set is information about individual feature-value pairs in the context of specific senses. Therefore, we can make the same independence assumption that has served us well in part-of-speech tagging, speech recognition, and probabilistic parsing — assume that the features are independent of one another. Making this assumption yields the following equation.

$$P(V|s) = \prod_{j=1}^{n} P(v_j|s) \tag{17.14}$$

Given this equation, *training* a Naive Bayes classifier amounts to collecting counts of the individual feature-value statistics with respect to each sense of the target word. The term $P(s)$ is the prior for each sense, which just corresponds to the proportion of each sense in the training set. Finally, since $P(V)$ is the same for all possible senses it does not effect the final ranking of senses, leaving us with the following.

$$\hat{s} = \underset{sinS}{\text{argmax}} P(s) \prod_{j=1}^{n} P(v_j|s) \tag{17.15}$$

Of course, all the issues discussed in Chapter 8 with respect to zero counts and smoothing apply here as well.

Rule		Sense
fish within window	⇒	bass^1
striped bass	⇒	bass^1
guitar within window	⇒	bass^2
bass player	⇒	bass^2
piano within window	⇒	bass^2
tenor within window	⇒	bass^2
sea bass	⇒	bass^1
play/V *bass*	⇒	bass^2
river within window	⇒	bass^1
violin within window	⇒	bass^2
salmon within window	⇒	bass^1
on bass	⇒	bass^2
bass are	⇒	bass^1

Figure 17.2 An abbreviated decision list for disambiguating the fish sense of bass from the music sense. (Adapted from (Yarowsky, 1996))

In a large experiment evaluating a number of supervised learning algorithms, Mooney (1996) reports that a naive-Bayes classifier and a neural network achieved the highest performance, both achieving around 73% correct in assigning one of 6 senses to a corpus of examples of the word *line*.

DECISION LIST CLASSIFIERS

Decision list classifiers can be viewed as a simplified variant of decision trees. In a decision list classifier, a sequence of tests is applied to each vector encoded input. If a test succeeds, then the sense associated with that test is applied to the input and returned. If the test fails, then the next test in the sequence is applied. This continues until the end of the list, where a default test simply returns the majority sense. Figure 17.2 shows a portion of a decision list for the task of discriminating the fish sense of *bass* from the music sense.

Learning a decision list classifier consists of creating a good sequence of tests based on the characteristics of the training data. There are wide number of methods that can be used to create such lists. Yarowsky (1994) employs an extremely simple technique that yields excellent results in this domain. In this approach, all possible feature-value pairs are used to create tests. These individual tests are then ordered according to their individual accuracy on the training set, where the accuracy of a test is based on its

log-likelihood ratio:

$$\text{Abs}(\text{Log}\left(\frac{P(Sense_1|f_i = v_j)}{P(Sense_2|f_i = v_j)}\right)) \tag{17.16}$$

The decision list is created from these tests by simplying ordering the tests in the list according to this measure, with each test returning the appropriate sense. Yarowsky (1996) reports that this technique consistently achieves over 95% correct on a wide variety of binary decision tasks.

We should note that this training method differs quite a bit from the standard decision list learning algorithm. For the details and theoretical motivation for that approach see (Rivest, 1987; Russell and Norvig, 1995).

Bootstrapping Approaches

BOOTSTRAPPING APPROACH

Not surprisingly, a major problem with supervised approaches is the need for a large sense-tagged training set. The **bootstrapping approach** (Hearst, 1991; Yarowsky, 1995) eliminates the need for a large training set by relying on a relatively small number of instances of each sense for each lexeme of interest. These labeled instances are used as *seeds* to train an initial classifier using any of the supervised learning methods mentioned in the last section. This initial classifier is then be used to extract a larger training set from the remaining untagged corpus. Repeating this process results in a series of classifiers with improving accuracy and coverage.

The key to this approach lies in its ability to create a larger training set from a small set of seeds. To succeed, it must include only those instances in which the initial classifier has a high degree of confidence. This larger training set is then used to create a new more accurate classifier with broader coverage. With each iteration of this process, the training corpus grows and the untagged corpus shrinks. As with most iterative methods, this process can be repeated until some sufficiently low error-rate on the training set is reached, or until no further examples from the untagged corpus are above threshold.

The initial seed set used in these bootstrapping methods can be generated in a number of ways. Hearst (1991) generates a seed set by hand labeling a small set of examples from the initial corpus. This approach has three major advantages:

- There is a reasonable certainty that the seed instances are correct, thus ensuring that the learner does not get off on the wrong foot
- The analyst can make some effort to choose examples that are not only correct, but in some sense prototypical of each sense.

Klucevsek **play**s Giulietti or Titano piano accordions with the more flexible, more difficult free **bass** rather than the traditional Stradella **bass** with its preset chords designed mainly for accompaniment.
We need more good teachers – right now, there are only a half a dozen who can **play** the free **bass** with ease.
An electric guitar and **bass play**er stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.
When the New Jersey Jazz Society, in a fund-raiser for the American Jazz Hall of Fame, honors this historic night next Saturday, Harry Goodman, Mr. Goodman's brother and **bass play**er at the original concert, will be in the audience with other family members.
The researchers said the worms spend part of their life cycle in such **fish** as Pacific salmon and striped **bass** and Pacific rockfish or snapper.
Associates describe Mr. Whitacre as a quiet, disciplined and assertive manager whose favorite form of escape is **bass fish**ing.
And it all started when **fish**ermen decided the striped **bass** in Lake Mead were too skinny.
Though still a far cry from the lake's record 52-pound **bass** of a decade ago, "you could fillet these **fish** again, and that made people very, very happy," Mr. Paulson says.
Saturday morning I arise at 8:30 and click on "America's best-known **fish**erman," giving advice on catching **bass** in cold weather from the seat of a bass boat in Louisiana.

Figure 17.3 Samples of *bass* sentences extracted from the WSJ using the simple correlates *play* and *fish*.

- It is reasonably easy to carry out.

A remarkably effective alternative technique is to simply search for sentences containing single words that are strongly correlated with the target senses. Yarowsky (1995) calls this the One Sense per Collocation constraint and presents results that show that it yields remarkably good results. For example, Figure 17.3 shows a partial result of a such a search for the strings "fish" and "play" in a corpus of *bass* examples drawn from the WSJ.

Yarowsky (1995) suggests two methods to select effective correlates: deriving them from machine readable dictionary entries, and selecting seeds using collocations statistics such as those described in Chapter 6. Putting all of this to the test, Yarowsky (1995) reports an average performance of 96.5% on a coarse binary sense assignment of 12 words.

Unsupervised Methods: Discovering Word Senses

Unsupervised approaches to sense disambiguation eschew the use of sense tagged data of any kind during training. In these approaches, feature-vector representations of unlabeled instances are taken as input and are then grouped into clusters according to a similarity metric. These clusters can then be represented as the average of their constituent feature-vectors, and labeled by hand with known word senses. Unseen feature-encoded instances can be classified by assigning them the word sense from the cluster to which they are closest according to the similarity metric.

Fortunately, clustering is a well-studied problem with a wide number of standard algorithms that can be applied to inputs structured as vectors of numerical values (Duda and Hart, 1973). The most frequently used technique in language applications is known as **agglomerative clustering**. In this technique, each of the N training instances is initially assigned to its own cluster. New clusters are then formed in a bottom-up fashion by successively merging the two clusters that are most similar. This process continues until a either a specified number of clusters is reached, or some global goodness measure among the clusters is achieved. In cases where the number of training instances makes this method too expensive, random sampling can be used on the original training set (Cutting *et al.*, 1992b) to achieve similar results.

AGGLOMERATIVE CLUSTERING

Of course, the fact that these unsupervised methods do not make use of hand-labeled data poses a number of challenges for evaluating the goodness of any clustering result. The following problems are among the most important ones that have to be addressed in unsupervised approaches.

- The correct senses of the instances used in the training data may not be known.
- The clusters are almost certainly heterogeneous with respect to the senses of the training instances contained within them.
- The number of clusters is almost always different from the number of senses of the target word being disambiguated.

Schütze's experiments (Schütze, 1992, 1998) constitute the most extensive application of unsupervised clustering to word sense disambiguation to date. Although the actual technique is quite involved, unsupervised agglomerative clustering is at the core of the method. As with the supervised approaches, the bulk of this work is directed at coarse binary distinctions. In this work, the first two problems are addressed through the use of pseudo-words and a hand-labeling of a small subset of the instances in each cluster.

The heterogeneity issue is addressed by assigning the majority sense to each of the induced clusters. Given this approach, the last problem is not an issue; the various discovered clusters are simply labeled with their majority sense. The fact that there may be multiple clusters with the same sense is not directly an issue in disambiguation.

Schütze's results indicate that for coarse binary distinctions, unsupervised techniques can achieve results approaching those of supervised and bootstrap methods. In most instances approaching the 90% range. As with most of the supervised methods, this method was tested on a small sample of words (10 pseudowords, and 10 real words).

Dictionary-Based Approaches

A major drawback with all of the approaches described above is the problem of scale. All require a considerable amount of work to create a classifier for each ambiguous entry in the lexicon. For this reason, most of the experiments with these methods report results ranging from 2 to 12 lexical items (The work of Ng and Lee (1996) is a notable exception reporting results disambiguating 121 nouns and 70 verbs). Scaling up any of these approaches to deal with all the ambiguous words in a language would be a large undertaking. Instead, attempts to perform large-scale disambiguation have focused on the use of **machine readable dictionaries**, of the kind discussed in Chapter 16. In this style of approach, the dictionary provides both the means for constructing a sense tagger, and the target senses to be used.

The first implementation of this approach is due to Lesk (1986). In this approach, all the sense definitions of the word to be disambiguated are retrieved from the dictionary. These senses are then compared to the dictionary definitions of all the remaining words in the context. The sense with the highest overlap with these context words is chosen as the correct sense. Note that the various sense definitions of the context words are simply lumped together in this approach. Lesk reports accuracies of 50-70% on short samples of text selected from Austen's *Pride and Prejudice* and an AP newswire article.

The problem with this approach is that dictionary entries for the various senses of target words are relatively short, and may not provide sufficient material to create adequate classifiers. [1] More specifically, the words used in the context and their definitions must have direct overlap with the words

[1] Indeed, Lesk (Lesk, 1986) notes that the performance of his system seems to roughly correlate with the length of the dictionary entries.

contained in the appropriate sense definition in order to be useful. One way to remedy this problem is to expand the list of words used in the classifier to include words related to, but not contained in their individual sense definitions. This can be accomplished by including words whose definitions make use of the target word. For example, the word *deposit* does not occur in the definition of *bank* in the American Heritage Dictionary (Morris, 1985). However, *bank* does occur in the definition of *deposit*. Therefore, the classifier for *bank* can be expanded to include *deposit* as a relevant feature.

Of course, just knowing that *deposit* is related to *bank* does not help much since we don't know to which of *bank's* senses it is related. Specifically, to make use of *deposit* as a feature we have to know which sense of *bank* was being used in its definition. Fortunately, many dictionaries and thesauri include tags known as subject codes in their entries that correspond roughly to broad conceptual categories. For example, the entry for *bank* in the *Longman's Dictionary of Contemporary English* (LDOCE) (Procter, 1978) includes the subject code EC (Economics) for the financial senses of *bank*. Given such subject codes, we can guess that expanded terms with the subject code EC will be related to this sense of bank rather than any of the others. Guthrie *et al.* (1991) report results ranging of 47% correct for fine-grained LDOCE distinctions to 72% for more coarse distinctions.

Note that none of these techniques actually exploit the dictionary entries *as definitions*. Rather, they can be viewed as variants of the supervised learning approach, where the content of the dictionary is used to provide the tagged training materials.

17.3 Information Retrieval

The field of information retrieval is of interest to us here due to its widespread adoption of word-based indexing and retrieval methods. Most current information retrieval systems are based on an extreme interpretation of the principle of compositional semantics. In these systems, the meaning of documents resides solely in the words that are contained within them. To revisit the Mad Hatter's quote from the beginning of Chapter 16, in these systems *I see what I eat* and *I eat what I see* mean precisely the same thing. The ordering and constituency of the words that make up the sentences that make up documents play no role in determining their meaning. Because they ignore syntactic information, these approaches are often referred to as **bag of words** methods.

BAG OF WORDS