Class-Based Statistical Models for Lexical Knowledge Acquisition

Stephen Clark

CSRP 539

September, 2001

ISSN 1350–3162

UNIVERSITY OF
SUSSEX
AT BRIGHTON

Cognitive Science
Research Papers

Acknowledgements

David Weir was an excellent supervisor, and was always available when I needed to discuss an idea with him. This availability even extended across the Channel, as we finished our first paper together over the telephone while David was on holiday. I also received good advice from my committee, Gerald Gazdar and Bill Keller, who were very helpful in steering the thesis in the right direction.

I would like to thank John Carroll for providing the training data for the empirical work, and for helpful suggestions regarding the work on parse selection. John never once complained when I asked him for the umpteenth time if he could send me more data. I have also benefited from discussions with Diana McCarthy, and I am particularly grateful to Diana for suggesting the pseudo disambiguation task as a possible evaluation, and for reading through a draft of the thesis. Darren Pearce helped enormously with the programming.

My examiners, Ted Briscoe and Geoff Sampson, devoted a substantial amount of time to reading the thesis, and their insightful comments were very helpful for turning the thesis into a technical report. I would also like to thank Ted for presenting a paper on David's and my behalf at NAACL 2001.

I am very grateful to Mark Steedman for allowing me time to complete the write-up in Edinburgh, and also to Mary McGee Wood for introducing me to NLP as a Masters student. And finally, special thanks to Naz Derakshan for providing the personal support.

Class-Based Statistical Models for Lexical Knowledge Acquisition

Stephen Clark

Abstract

This thesis is about the automatic acquisition of a particular kind of lexical knowledge, namely the knowledge of which noun senses can fill the argument slots of predicates. The knowledge is represented using probabilities, which agrees with the intuition that there are no absolute constraints on the arguments of predicates, but that the constraints are satisfied to a certain degree; thus the problem of knowledge acquisition becomes the problem of probability estimation from corpus data. The problem with defining a probability model in terms of senses is that this involves a huge number of parameters, which results in a sparse data problem. The proposal here is to define a probability model over senses in a semantic hierarchy, and exploit the fact that senses can be grouped into classes consisting of semantically similar senses.

A novel class-based estimation technique is developed, together with a procedure that determines a suitable class for a sense (given a predicate and argument position). The problem of determining a suitable class can be thought of as finding a suitable level of generalisation in the hierarchy. The generalisation procedure uses a statistical test to locate areas consisting of semantically similar senses, and, as well as being used for probability estimation, is also employed as part of a re-estimation algorithm for estimating sense frequencies from incomplete data.

The rest of the thesis considers how the lexical knowledge can be used to resolve structural ambiguities, and provides empirical evaluations. The estimation techniques are first integrated into a parse selection system, using a probabilistic dependency model to rank the alternative parses for a sentence. Then, a PP-attachment task is used to provide an evaluation which is more focussed on the class-based estimation technique, and, finally, a pseudo disambiguation task is used to compare the estimation technique with alternative approaches.

Submitted for the degree of D. Phil.

University of Sussex

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is about the automatic acquisition of a particular kind of lexical knowledge, namely the knowledge of which noun senses can fill the argument slots of predicates. Knowledge of this kind is closely related to the classical notion of selectional restrictions (Katz and Fodor 1964) and selectional preferences (Wilks 1975; Resnik 1993a). However, there is a difference, in that selectional restrictions (and preferences) are usually expressed as constraints on the semantic class of an argument; a much used example is that the verb *drink* constrains its object to be a kind of liquid (or the verb *drink* 'strongly prefers' a kind of liquid). The purpose of this thesis is not to acquire a set of classes that best represent the selectional preferences of a predicate, as others have done (Resnik 1993a; Ribas 1995b; Wagner 2000), but rather to consider how well an *individual sense* satisfies the preferences of a predicate.

In order to avoid confusion, then, we do not refer to the lexical knowledge in question as selectional restrictions or selectional preferences, but use the term *lexical sense preferences*. The terms *selectional restrictions* and *selectional preferences* will be used to refer to the constraints a predicate places on the semantic class of its arguments. We may also use just use the generic term *preferences* to refer to selectional, or lexical sense, preferences. The next section gives some motivation for acquiring lexical sense preferences.

1.1 Uses of lexical sense preferences

Knowledge of lexical sense preferences is useful for a variety of NLP tasks, such as structural disambiguation, parsing, word sense disambiguation, anaphora resolution and language modelling. To see how such knowledge can be used to resolve structural ambiguities, consider this example from Charniak 1993:

(1.1) Fred awarded a prize for the dog that ran the fastest.

This is an example of a relative clause attachment ambiguity, since the relative clause *that ran the fastest* can attach to either *prize* or *dog*. To resolve the ambiguity, we can use the fact that the correct sense of *dog* is more likely to be a subject of *run* than the correct sense of *prize*. Another form of structural ambiguity is prepositional phrase (PP) attachment ambiguity; consider the following example, in which the PP *with a spoon* attaches to either *ate* or *strawberries*:

(1.2) Fred ate strawberries with a spoon.

If we think of *ate-with* and *strawberries-with* as predicates, then this ambiguity can be resolved by noting that *ate-with* is more likely to take the correct sense of *spoon* as an argument than *strawberries-with*.

This basic approach can be applied to other problems, such as anaphora resolution and word sense disambiguation. Consider the problem of determining the referent of *it* in the following sentence, taken from Wilks 1975:

(1.3) I bought the wine, sat on a rock and drank it.

To determine the correct referent, we can use the fact that the correct sense of *wine* is more likely to be an object of *drank* than the correct sense of *rock*. Word sense disambiguation can be tackled in a similar way; in the sentence *Mary drank burgundy* (Resnik 1997), the correct sense of *burgundy* can be determined from the fact that the beverage sense is more likely to be an object of *drank* than the colour sense.

As a final application, consider the following example from language modelling for speech recognition. The problem is to decide which of the following two strings is the most likely, assuming that both could have given rise to a similar acoustic signal:

(1.4) The dogs barked.

(1.5) The dog sparked.

The problem can be resolved by noting that the correct sense of *dog* is more likely to be a subject of *barked* than *sparked*.

Some of these examples may appear a little contrived (as examples often are), and, in fact, one of the conclusions of this thesis will be that lexical sense preferences alone cannot always resolve the ambiguity. For example, if 1.1 had been *Fred awarded a* cat *for the dog that ran the fastest*, the preferences of *run* for its subject would not have been able to discriminate between the two attachment points. However, it is widely believed that lexical sense preferences are a useful source of knowledge for resolving ambiguities of various kinds. To give one example, the current best performing wide-coverage statistical parsers are all based on lexicalised models of some form (Collins 1996, 1997; Ratnaparkhi 1999; Charniak 1997, 2000), and Collins (1996) notes that the dependency relations that are central to his parser "can also be viewed as representing a semantic predicate-argument relationship, with the three elements [of the triple denoting the relation] being the type of the argument, result and functor respectively." Thus lexical sense preferences are very similar to the dependencies underlying Collins' parsing model.

One difference between Collins' dependencies and lexical sense preferences is that Collins' dependencies are between nouns, rather than noun senses, which raises the question of why we are choosing to focus on senses. Indeed, each of the previous examples of ambiguity (with the exception of the word sense disambiguation example) could have been resolved by considering the nouns themselves. For example, 1.1 could be resolved by considering the probabilities of the words *dog* and *prize* appearing as the subject of *run*, since we would expect the probability of *dog* appearing as a subject of *run* to be greater than the corresponding probability for *prize*.

One reason for using senses is that the relations we are trying to capture are semantic in nature, and the success of lexical approaches to problems such as structural disambiguation is arguably due to the fact that words provide a 'poor man's semantics'. The reason that nouns can be used to resolve the relative clause ambiguity is because of the underlying semantics: dogs run, on the whole, whereas prizes do not. Thus it might be thought that using noun senses, which are closer to the underlying semantics, would improve performance on such tasks. In fact, this is yet to be demonstrated, although there is some anecdotal evidence. For example, the current best performing classifier on the PP-attachment task (Stetina and Nagao 1997) uses the level of association between senses from a semantic hierarchy to resolve PP-attachment ambiguities.

Another reason for using noun senses is that they provide a direct route into a semantic hierarchy, which can be used to aid the acquisition process. In the next section we motivate the use of probabilities to represent lexical sense preferences, and in Section 1.3 we show how the classes from a semantic hierarchy can be used to help estimate these probabilities.

1.2 Using probabilities to represent preferences

Resnik (1993a) argues that the constraints a predicate places on its arguments are not Boolean constraints, as in the classical account of selectional restrictions (Katz and Fodor 1964), but that the constraints are satisfied to a certain degree. (Resnik cites McCawley (1968) and Fodor (1977) as earlier critics of Katz and Fodor's theory.) We follow Resnik in modelling the constraints as graded preferences, and, in line with other recent work in this area (Ribas 1995b; Li and Abe 1998; McCarthy 2000; Wagner 2000), probabilities are used to encode the preferences. An important question is whether the preference measure should define a probability distribution over the possible arguments of a predicate.

Resnik's measure of selectional preference, which he calls 'selectional association,' is defined in terms of probabilities, but the measure does not define a probability distribution over the possible arguments of a predicate; the values for selectional association need not lie between zero and one, and do not sum to one over the possible arguments. This is also true of a number of related measures in the literature, such as the chi-squared statistic (Kilgarriff 1996), likelihood ratio statistics (Dunning 1993) and mutual information (Church and Hanks 1990). Aside from the question of whether these measures are appropriate for use in corpus-based linguistics (Dunning 1993), they all suffer from a limitation.

The limitation arises when determining the 'semantic plausibility' of a complex linguistic event, such as a parse tree. In order to do parse selection, one can measure the overall extent to which the arguments in a parse satisfy the preferences of their predicates; if a parse has a number of semantically implausible arguments, this is an indication that the parse is incorrect. However, a difficulty arises in combining individual preference scores to determine an overall score for the parse. If the preferences are measured using selectional association, or any of the other measures mentioned above, then there is no established theory of how to combine the scores. This problem does not arise for a probability distribution, since there is a well established theory, namely probability theory, which deals with the problem of how to combine probabilities.

Another advantage of using probabilities is that additional knowledge can be integrated into a probability model containing the preferences. An important factor in resolving relative clause attachments, for example, is that the correct attachment is often the closest possible attachment to the clause (Cardie 1992). To decide on the referents of anaphors, Ge, Hale, and Charniak (1998) use a measure of selectional preference, but in conjunction with a number of different knowledge sources. For word sense disambiguation, Resnik (1997) notes that preference information is not always enough, since some verbs and modifiers do not have a strong preference for their arguments, and an important source of additional information is the surrounding context. And in parsing, structural information is important; Collins (1999) notes that, in English, there is a preference for right branching structures. In all of these cases, if the various knowledge sources can be represented using probabilities, then there is a well understood framework that can be used to combine the different sources.

For these reasons, probabilities are used to represent lexical sense preferences. The probabilities are of the form $p(c|v,r)$, where c is a noun sense, v is a predicate and r is an argument position. The interpretations of the terms 'noun sense,' 'predicate' and 'argument position,' as used in this thesis, will be given in Chapter 3, where the estimation problem is precisely defined. The next section considers the problem of estimating the probabilities.

1.3 Estimating probabilities of senses

A feature of the estimation problem, in line with many other problems in statistical NLP, is that there are many probabilities to be estimated. The noun senses that are considered here are taken from a semantic hierarchy containing over $60,000$ senses, and we are required to estimate a probability for each of these senses, for every argument position of every predicate. Clearly, such a large number of parameters results in a serious sparse data problem. In order to reduce the number of parameters, we propose to define a probability model over senses in a semantic hierarchy, and exploit the fact that senses can be grouped into classes consisting of semantically similar senses. The assumption underlying this approach is that the probability of a noun sense can be approximated by a probability based on a suitably chosen class. For example, it seems reasonable to suppose that the probability of (the food sense of) *chicken* appearing as an object of the verb *eat* can be approximated in some way by a probability based on a class such as FOOD. There are two elements to the problem of using a class to estimate the probability of a noun sense. First, given a suitably chosen class, how can that class be used to estimate the probability of the sense? And second, given a noun sense, how can a suitable class be determined?

The semantic classes that are used are from the noun hierarchy in WordNet (Fellbaum 1998b), and thus the second question can be rephrased as how to use a WordNet class to determine the probability of an individual sense. A novel solution to this question is given in Chapter 3. A central concern of this thesis is the first question, which can be thought of as how to determine a suitable level of generalisation in the noun hierarchy. The novel solution that is developed uses a statistical test to determine whether an estimate based on a particular class is likely to be a good estimate of the probability of the sense. The test achieves this by determining whether a class consists of semantically similar senses, or is a ‘homogeneous’ set of senses.

An important point relating to the generalisation problem is that we are not trying to acquire selectional preferences in the way this is often construed (Resnik 1993a; Ribas 1995b; Wagner 2000). The problem being considered is that of probability estimation, and the point of the generalisation is to determine a class that can help with the sparse data problem. Consider determining a level for ⟨beefburger⟩ in the object position of *eat*; if the procedure returns ⟨snack_food⟩, say, rather than the ‘more intuitive’ ⟨food⟩, this should not necessarily be considered a failure. It may be that the class SNACK_FOOD leads to a more accurate estimate.[1] This issue will be discussed further in Chapter 3.

1.4 Overview of chapters

Chapter 2 describes related work, and there is a particular focus on class-based statistical methods using WordNet. Other approaches to probability estimation are also described, and we consider some of the statistics that have been used for lexical acquisition. Finally, we look at some applications that have used lexical knowledge similar to the kind we are considering. There is an emphasis on statistical parsing and the resolution of PP-attachment ambiguities, since these two applications are considered later in the thesis.

Chapter 3 describes the main estimation method central to the thesis, which is based on novel solutions to the following problems: how to use a class to estimate the probability of a sense, and how to find a suitable class, or level of generalisation, in the hierarchy. Issues relating to the use of a chi-squared statistic, which is used as part of the generalisation procedure, are addressed. We also show how the level of generalisation varies with changes in the sample size and the level of significance used in the chi-squared test.

Chapter 4 considers the word sense disambiguation problem for training. The problem is to determine the number of times a noun sense appears as an argument to a predicate, assuming the data are not sense disambiguated. An iterative procedure is developed which re-estimates the frequencies, starting by simply splitting the count for a noun equally among the noun’s senses. A feature of the re-estimation algorithm is that it uses the generalisation procedure developed in Chapter 3, and this leads to a new interpretation of the procedure in terms of finding homogeneous sets of senses in the hierarchy.

Chapter 5 shows how the estimation techniques can be integrated into a parse selection system. The system uses a probabilistic dependency model, based on an inventory of grammatical

[1] Nouns in upper case will sometimes be used to denote classes. An alternative notation for classes will be introduced in Chapter 3.

relations, to select a parse. The work in this chapter extends similar work using a semantic hierarchy, which has only looked at particular ambiguities in isolation, such as PP-attachment and noun-noun compound ambiguities (Li and Abe 1998; Resnik 1998).

Chapter 6 gives two further evaluations that are more focussed on the estimation techniques. The first evaluation uses a PP-attachment task, and demonstrates that the generalisation procedure outperforms a simple alternative of using a fixed level of generalisation. The second evaluation uses a pseudo disambiguation task to compare the class-based estimation technique with alternatives based on work by Resnik (1993a) and Li and Abe (1998). As well as giving positive results, the evaluation provides a novel result regarding the use of the chi-squared statistic.

Chapter 7 outlines the contributions of the thesis, and considers possibilities for future work.

Chapter 2

Previous Work

This chapter is divided into two sections; one section describes work from those areas of lexical acquisition that are of particular relevance to this thesis, and the other section describes previous approaches to structural disambiguation and parse selection. These areas of application are considered because the problems of structural disambiguation and parse selection are dealt with in Chapters 5 and 6.

The knowledge acquisition section focuses on selectional preferences, describing in detail those approaches that have used WordNet and showing how they relate to the class-based estimation method described in Chapter 3. We also describe some approaches to automatic clustering, which is an important alternative to using a man-made hierarchy for generalisation, and also collocation extraction, which has used statistics that are used in Chapters 3 and 4. Finally, a number of smoothing techniques for probability estimation are described; this work is relevant because the class-based estimation method described in Chapter 3 can be thought of as performing a kind of smoothing.

The applications section focuses on those approaches to structural disambiguation and parse selection that have used knowledge similar to lexical sense preferences; this includes much of the recent work on resolving PP-attachment ambiguities and statistical parsing, where there has been a move towards probability models based on lexical dependencies.

2.1 Lexical knowledge acquisition

The role of the lexicon has taken on increasing importance in recent years, both from a theoretical and a computational perspective. On the theoretical side, many grammatical formalisms now use the lexicon to encode much of the structural and semantic information needed for describing the sentences of a language. Examples of such formalisms include Lexical-Functional Grammar (Bresnan and Kaplan 1982), Head-Driven Phrase Structure Grammar (Pollard and Sag 1994), Tree-Adjoining Grammar (Joshi and Schabes 1992), and Combinatory Categorial Grammar (Steedman 2000). On the computational and applied side, there has been a move away from 'toy' NLP systems with small lexicons, to systems that can handle a wide variety of naturally occurring text; such systems generally require a substantial lexicon with many thousands, or tens of thousands, of entries. Thus the acquisition of lexical knowledge has become an important topic in NLP (Boguraev and Pustejovsky 1996).

Early approaches to building NLP lexicons consisted of either building the lexicon manually or deriving the lexicon from machine-readable dictionaries (see, for example, Boguraev, Briscoe, Carroll, Carter, and Grover 1987; Grishman, Macleod, and Meyers 1994). However, as Briscoe and Carroll (1997) point out, neither approach can lead to a comprehensive or completely accurate lexicon. The problem with both approaches is that they rely on the manual efforts of linguists, which are limited in the following ways:

1. It is very time consuming to create a large lexicon by hand.

2. The words in a language are constantly changing, and so it is difficult to keep the lexicon up to date.

3. Some types of lexical information, particularly quantitative information, are difficult to collect manually (Manning and Schütze 1999, Ch.8).

Automatic approaches that derive lexical knowledge from corpora are able to address all of these limitations. In response to 1, automatic approaches can collect large amounts of information very quickly, which not only reduces the initial cost of creating a lexicon, but also means that a lexicon can be quickly adapted to a new domain. For 2, automatic approaches can keep the lexicon up to date, by using the most up to date corpora. And for 3, automatic approaches can not only cope with the sheer volume of quantitative information that may be required, but can also estimate the relevant quantities in an objective fashion. This thesis provides a good example of a problem where very large amounts of quantitative information are being acquired, since probabilities are being estimated for every combination of predicate, noun sense, and argument position. A further reason for not adopting a manual approach in this thesis is that it would be very difficult to subjectively estimate the probabilities of lexical sense preferences.

There are also disadvantages to acquiring lexical knowledge automatically from corpora. Any corpus is likely to contain anomalies, which will be reflected in the acquired knowledge. In addition, the system used for acquisition is unlikely to be completely reliable, since automatic lexical acquisition can be a difficult task. However, the advantages of automatic acquisition for building large lexicons greatly outweigh these disadvantages.

2.1.1 Acquisition of selectional preferences

The acquisition of selectional preferences is the area of knowledge acquisition that relates most closely to this thesis. We begin with a brief history and then describe the approaches to acquisition that have used a man-made hierarchy as prior knowledge.

Selectional preferences are the constraints that a predicate places on the semantic type of its arguments. Such constraints, in the form of graded preferences or Boolean restrictions, have proven useful in building NLP systems and tackling various tasks; Allen (1995) goes as far to say that "Selectional restrictions … are used in some form in almost every computational [NLP] system." To give one example of a system, the Core Language Engine (Alshawi 1992) uses sortal restrictions, defined as "constraints on the sorts of objects that can fill argument positions of specified relations", to rule out incorrect interpretations at the logical form level. Alshawi 1992 contains the following example:

(2.1) Trinity was built by a river.

The fact that a river is not the kind of object that can build things can be used to rule out the interpretation in which the river is the agent of the building event.

The classical notion of a selectional restriction is due to Katz and Fodor (1964), in which the arguments of a predicate are required to be of a certain semantic type. A much used example is that *drink* constrains its object to be a kind of liquid. However, it is not difficult to find legitimate examples, often arising from metaphorical or fictional usage, in which selectional restrictions are broken. These examples, such as cars drinking gasoline and people eating words and hats, are not unusual or rare.

The existence of such cases led Wilks (Wilks 1975; Wilks and Fass 1992) to propose the notion of a *semantic preference*, in which a predicate does not restrict the semantic type of its arguments, but rather has a *preferred* kind of argument. However, Wilks distanced himself from a probabilistic treatment of preferences: it is still the case that an individual preference is either satisfied or it is not, as with selectional restrictions. The difference is that an interpretation of a sentence can be preferred, even if individual preferences are violated, as long as there is no alternative interpretation with less violations.

Resnik (1993a) took the notion of preference one step further, by suggesting that preference should be measured on a continuous scale. Resnik uses the following list of examples (which originally appeared in Drange 1966) to demonstrate that the preferences of *like coffee* for its argument can be satisfied to a certain degree:

(2.2) Englishmen like coffee better than tea.

Squirrels like coffee better than tea.

Protozoa like coffee better than tea.

Bacteria like coffee better than tea.

Milkweed plants like coffee better than tea.

Stones like coffee better than tea.

Electrons like coffee better than tea.

Quadratic equations like coffee better than tea.

The list does show a general trend, with the perfectly acceptable *Englishmen* at one end of the preference scale and the semantically bizarre *Quadratic equations* at the other.

As far as we are aware, Resnik was the first person to formulate a statistical model of selectional restrictions or preferences. Resnik's model is based on classes from a semantic hierarchy, and much of the recent work in acquiring selectional preferences, particularly the work that uses a man-made hierarchy as prior knowledge, has been motivated by Resnik's approach. We describe this work below, after describing the hierarchy that is typically used.

The most widely used man-made hierarchy is the noun hierarchy in WordNet (Fellbaum 1998b). The hierarchy consists of noun senses, or *concepts*, related by the 'is-a-kind-of' relation. One of the difficulties in describing the various work using WordNet is that different researchers use terms like *concept* in different ways. Rather than modify the terminology for each piece of work, we aim for consistency throughout the thesis. Following Miller (1998), we use *lexicalised concept*, or just *concept*, to refer to a noun sense; in particular, *concept* is *not* used to refer to a set of senses. A concept is represented by a *synset*, which is the set of synonymous words that can be used to denote that concept. For example, the synset for the concept $\langle\texttt{aeroplane}\rangle$ is $\{$*airplane*, *aeroplane*, *plane*$\}$. Unless stated otherwise, we use *class* to refer to a set of concepts, such that each concept in the hierarchy has a corresponding class (consisting of the concept itself and those dominated by the concept). The class AEROPLANE contains all the concepts dominated by $\langle\texttt{aeroplane}\rangle$ (including $\langle\texttt{aeroplane}\rangle$ itself): $\langle\texttt{aeroplane}\rangle$, $\langle\texttt{airliner}\rangle$, $\langle\texttt{biplane}\rangle$, $\langle\texttt{bomber}\rangle$, $\langle\texttt{airbus}\rangle$, and so on. Two further terms we will use are *hypernym* and *hyponym*: a concept c' is a hypernym of c, and c a hyponym of c', if c is-a-kind-of c'.[1] This description of the hierarchy should suffice for this chapter; a more precise description is given in Chapter 3.

[1] Strictly speaking, the correct use of the Greek root *onym* would produce the term *hyperonym*, and not *hypernym*. However, we follow standard usage in adopting the latter term. Thanks to Geoff Sampson for pointing this out.

Resnik's model of selectional preference

The parts of Resnik's work (1993a, 1993b, 1996, 1997, 1998, 1999a, 1999b) that are most relevant for this thesis are his solutions to the following questions:

1. How can a probability distribution over the WordNet hierarchy be defined?[2]
2. How can we measure the extent to which an argument satisfies the preferences of a predicate?

Each question will be dealt with in turn.

Resnik defines his probability model in terms of classes (where *class* has the interpretation given above). Let $C = \{c_1, c_2, \ldots, c_k\}$ be the set of classes in WordNet, where k is the number of concepts (so that each concept has a corresponding class). Resnik places the following constraints on any probability distribution over C:[3]

$$\text{if } c_i \text{ is-a-kind-of } c_j \text{ then } p(c_j) \geq p(c_i) \tag{2.3}$$

$$\textstyle\sum_{i=1}^{k} p(c_i) = 1 \tag{2.4}$$

Equation 2.3 agrees with the intuition that the probability of a class increases with the level of abstraction. (Although note that the probability corresponding to a node in the hierarchy is not defined in terms of the sum of the probabilities of the children.) Equation 2.4 is required by Resnik because he defines a random variable ranging over all the classes, and defines information-theoretic functions of that random variable such as entropy.

Resnik's aim is to model the fact that some verbs select more strongly for their arguments than others. For example, *eat* selects more strongly for its direct object than *find*. Resnik's approach is based on the fact that, for strongly selecting verbs, the probability of a class conditional on the verb, $p(c|v)$, is likely to differ largely from the unconditional probability, $p(c)$. From an information-theoretic perspective, a strongly selecting verb provides more information about the class of its argument than a weakly selecting verb. This idea led Resnik to define the *selectional preference strength* of a verb (assuming some argument position) as follows:

$$S_R(v) = \mathrm{D}(p(c|v) \| p(c)) = \sum_{c \in C} p(c|v) \log \frac{p(c|v)}{p(c)} \tag{2.5}$$

The quantity $\mathrm{D}(p(c|v)\|p(c))$ is the *Kullback-Leibler divergence*, or *relative entropy*, between the two probability distributions, and is an information-theoretic measure of the 'distance' between two distributions (Cover and Thomas 1991). Note that the measure takes into account the difference between the posterior and prior probabilities for *all* the classes in the hierarchy.

Resnik's next step is to suggest a measure of how well a *particular* class satisfies the preferences of a verb, which he calls *selectional association*:

$$A_R(v,c) = \frac{1}{S_R(v)}\, p(c|v) \log \frac{p(c|v)}{p(c)} \tag{2.6}$$

$A_R(v,c)$ is the contribution that class c makes to the overall preference strength of v, normalised by the overall strength. Note that the selectional association is not a probability, which means it cannot be integrated directly into a probability model for use in parsing or some other application. It was argued in Chapter 1 that this is a disadvantage of the measure, which partly motivated the use of probabilities to represent lexical sense preferences.

[2] There are other taxonomies in WordNet, but a reference here to the WordNet hierarchy refers to the noun hierarchy only.

[3] Strictly speaking, the is-a-kind-of relation should be between the concepts corresponding to the classes, rather than the classes themselves, but Resnik does not make a clear distinction between the two.

A difficulty with using selectional association in an application is that the arguments are likely to be nouns, rather than classes, and so an appropriate class has to be chosen for the noun. This problem has two dimensions, since a noun can have more than one sense, but can also be represented at various levels of abstraction. Resnik neatly refers to these two dimensions as 'horizontal' and 'vertical' ambiguity, respectively. Resnik's suggestion is to address both problems in one step, by choosing the class that maximises the selectional association score. Resnik 1998 contains the example of *letter* appearing in the object position of *write*. For the version of WordNet being used, *letter* has three senses, and is a member of 19 classes overall, taking into account all levels of abstraction.[4] From these classes, Resnik found that the class with the highest estimated association score (based on data from the Brown Corpus) was ⟨`writing`⟩ (anything expressed in letters; reading matter). This appears to be a suitable solution to the vertical ambiguity problem, and, since the only sense of *letter* that is dominated by ⟨`writing`⟩ is the 'written message' sense, the problem of horizontal ambiguity appears to have been solved as well. Resnik (1997) considers whether this approach can be applied generally to the problem of word sense disambiguation.

The vertical ambiguity problem is a central concern of this thesis. Regarding Resnik's proposed solution, Li and Abe (1998) comment that "This method is based on an interesting intuition, but its interpretation as a method of estimation is not clear." One of the aims of this thesis has been to find a solution to the vertical ambiguity problem that has a clearer statistical interpretation.

There is another problem with Resnik's solution, in that it does not always generalise appropriately for arguments that are negatively associated with a predicate. To see why, consider the problem of deciding how well ⟨`location`⟩ satisfies the preferences of *eat*. Since locations are not the kinds of things that are typically eaten, a suitable level of generalisation would be a class that has a low selectional association with respect to *eat*. However, ⟨`location`⟩ is a kind of ⟨`entity`⟩ in WordNet,[5] and choosing the class with the highest selectional association score is likely to produce ⟨`entity`⟩ as the level of generalisation. This is a problem, because the selectional association of ⟨`entity`⟩ with respect to *eat* will probably be too high to reflect the fact that ⟨`location`⟩ is a very unlikely object of the verb. The solution to the vertical ambiguity problem presented in Chapter 3 is able to generalise appropriately in such cases.

Resnik's approach to probability estimation is straightforward, using a relative frequency estimate for the probability of a class (the equations given here are for unconditional probabilities, but it is trivial to extend them to probabilities conditional on a verb and argument position):

$$\hat{p}(c) = \frac{\mathrm{freq}(c)}{\sum_{c' \in C} \mathrm{freq}(c')} \tag{2.7}$$

where $\mathrm{freq}(c')$ is the number of times class c' appears in the data. The difficulty is in estimating $\mathrm{freq}(c')$, since the data are assumed to be nouns, and the problems of both horizontal and vertical ambiguity arise. Resnik adopts a simple approach, by distributing the count for a noun equally among *all* the classes of which the noun is a member:

$$\mathrm{freq}(c) = \sum_{w \in words(c)} \frac{1}{|classes(w)|} \mathrm{freq}(w) \tag{2.8}$$

where *words(c)* is the set of words in the synsets of concepts dominated by c (including c itself), and *classes(w)* is the set $\{c \mid w \in words(c)\}$. To make this clear, consider the following example. The noun *wine* has two senses in WordNet, a colour sense and a beverage sense; the beverage sense is dominated by 11 concepts, and the colour sense by 8 concepts, which means that *wine* is a member of 21 classes overall (including the classes containing just the individual concepts): WINE, DARK_RED, RED, CHROMATIC_COLOR, COLOR, ..., ABSTRACTION; WINE,

[4] We have defined classes as sets of *concepts*. However, Resnik also has the notion of a class containing nouns, so that a class can be thought of as containing all those nouns that are in the synsets of the concepts belonging to the class.

[5] For example, the hypernyms of the concept ⟨`Dallas`⟩ are as follows: ⟨`city`⟩, ⟨`municipality`⟩, ⟨`urban_area`⟩, ⟨`geographical_area`⟩, ⟨`region`⟩, ⟨`location`⟩, ⟨`object`⟩, ⟨`entity`⟩.

BEVERAGE, FOOD, LIQUID, FLUID, ..., ENTITY. Each of these classes would receive a count of $1/21$ for each instance of *wine* in the data. Note that this method of class estimation is unusual among the work in this area, and is motivated by the desire to define a probability distribution over the set of all classes. The other work described here does not distribute the count in this way, and does not define models that satisfy equation 2.4.

Resnik (1993a) uses his model of selectional preference to predict which verbs can 'drop' their direct objects, the idea being that verbs with higher preference strengths are able to drop their objects more readily. He also applies the model to resolving various forms of structural ambiguity; this part of the work will be described in Section 2.2.1, along with other approaches to structural disambiguation.

Extensions to Resnik's approach

Resnik's techniques were adopted, and in some ways modified, by Ribas (1994, 1995a, 1995b). Much of Ribas's work consists of various modifications to the basic approach, but one proposal worthy of particular consideration is the following estimate for the frequency of a class:

$$\mathrm{freq}(c) = \sum_{w \in words(c)} \frac{|senses(w) \in c|}{|senses(w)|} \mathrm{freq}(w) \tag{2.9}$$

where *senses(w)* is the set of senses of *w*, and *words(c)* is defined as before; $|senses(w) \in c|$ is the number of senses of *w* that are in class *c*. To make this clear, consider the wine example adapted to Ribas' scheme: the two senses of *wine* receive a count of $1/2$ for each instance of *wine* in the data. In addition, each of these counts is 'passed up' the hierarchy, so that each ancestor of the two senses also receives a count of $1/2$. The probability of a class is then estimated as follows:

$$\hat{p}(c) = \frac{\mathrm{freq}(c)}{N} \tag{2.10}$$

where *N* is the number of noun tokens in the data. This approach to probability estimation results in the following distribution over classes:

$$\hat{p}(\text{ROOT}) = 1 \tag{2.11}$$

where ROOT is the class containing all the concepts. Much of the work estimating probabilities over WordNet uses this method of distributing the count for a noun, including the class-based estimation technique described in Chapter 3. The precise description of the probabilities of senses and classes given in that chapter should make it clear why we have adopted Ribas' scheme.

Ribas' ultimate aim is to find a set of classes which best represents the selectional preferences of a verb. He does this by conducting a search through the space of classes, and choosing those classes that maximise the score in 2.6, subject to the constraint that the final classes are mutually disjoint. (The probabilities in 2.6 are estimated according to Ribas' scheme, and the term $S_R(v)$ is ignored, since it is a constant for any particular verb and argument position.) The work by Li and Abe (Li 1998; Li and Abe 1998; Abe and Li 1996) has a similar goal, but has a stronger theoretical foundation in that it employs a well known learning technique from the field of machine learning. This work is described next.

Using MDL to select a level of abstraction

Li and Abe employ the Minimum Description Length (MDL) principle to select a set of classes from a hierarchy, together with their associated probabilities, to represent the selectional preferences of a verb. The preferences and class-based probabilities are then used to estimate the probability distribution $p(n|v,r)$, where *n* is a noun, *v* is a verb and *r* is an argument slot. These probabilities represent what Li and Abe call *case frame patterns*. Note that the acquisition problem Li and Abe address is very similar to the acquisition of lexical sense preferences.

Li and Abe's application of MDL requires the hierarchy to be in the form of a thesaurus, where each leaf node represents a noun, and internal nodes represent the class of nouns that the

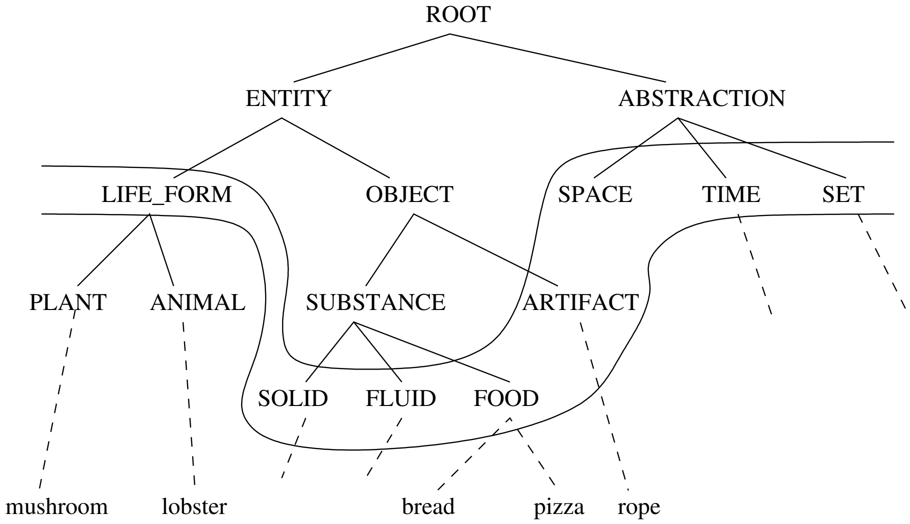

Figure 2.1: Example thesaurus and cut

node dominates. (For a thesaurus defined in this way, we use *class* to refer to a set of nouns.) The hierarchy is also assumed to be in the form of a tree. The class-based models consist of a partition of the set of nouns (leaf nodes) and a probability associated with each class in the partition. The probabilities are the conditional probabilities of each class, given the relevant verb and argument position. Li and Abe refer to such a partition as a 'cut', and the cut together with the probabilities, a 'tree cut model'. The probabilities of the classes in a cut, Γ, satisfy the following constraint:

$$\sum_{C \in \Gamma} p(C|v,r) = 1 \tag{2.12}$$

In order to determine the probability of a noun, the probability of a class is assumed to be distributed uniformly among the members of that class:

$$p(n|v,r) = \frac{1}{|C|}\, p(C|v,r) \quad \text{for all } \; n \in C \tag{2.13}$$

A simplified thesaurus is shown in Figure 2.1, together with an example cut for the object position of *eat* (based on an example from Li and Abe 1998). Since the class in the cut containing *pizza* is FOOD, the probability $p(pizza|eat,\text{obj})$ would be estimated as $p(\text{FOOD}|\text{eat},\text{obj})/|\text{FOOD}|$.

The uniform distribution assumption (2.13) means that cuts close to the root of the hierarchy result in a greater smoothing of the probability estimates than cuts near to the leaves. Thus there is a trade-off between choosing a model that has a cut near the leaves, which is likely to overfit the data, and a more general (simple) model near the root, which is likely to underfit the data. MDL looks ideally suited to the task of model selection, since it is designed to deal with precisely this trade-off. The simplicity of a model is measured using the *model description length*, which is an information-theoretic term and denotes the number of bits required to encode the model. The fit to the data is measured using the *data description length*, which is the number of bits required to encode the data (relative to the model). The overall description length is the sum of the model description length and the data description length, and the MDL principle is to select the model with the shortest description length.

We will not go into the details of how Li and Abe carry out the encoding, but to give more of the intuition behind the approach, consider this explanation from Li (1998) (p.52):

> The MDL-based method, in fact, conducts generalization in the following way. When the differences between the frequencies of the words in a class are not large enough

> (relative to the entire data size and the number of words), it generalizes them into a class. When the differences are especially noticeable (relative to the entire data size and the number of the words), on the other hand, it stops generalization at that level.

As we shall see, a similar approach to generalization is taken in this thesis (but not using MDL).

One of the problems with this generalization approach is that it is based on frequencies, which are not always a good measure of which nouns should be generalised into a class. For example, *bread* and *artichoke* may have very different frequencies (with respect to the object position of *eat*), simply because *bread* is more likely to appear in a corpus than *artichoke*. Another way to think of this is that the prior probability of bread ($p(bread|\mathtt{obj})$) is higher than the prior probability of *artichoke* ($p(artichoke|\mathtt{obj})$). This idea led Abe and Li (1996) to try and incorporate the prior probability into the MDL generalisation process. However, this work does not apply MDL in such a theoretically sound way; as Li (1998) puts it, "See (Abe and Li 1996) for a *heuristic method* for learning a similar measure on the basis of the MDL principle." (p.42, emphasis added) Note that the generalisation procedure described in Chapter 3 does not suffer from this problem of comparing frequencies, since the decision to group senses into a class is not based purely on the frequencies of a sense.

In practice, there are a number of problems in applying the MDL approach to the WordNet hierarchy:

1. WordNet is a hierarchy of noun *senses*.

2. Many of the nouns appearing in synsets are represented at internal nodes, rather than the leaves.

3. WordNet is a DAG, not a tree.[6]

The first two problems arise from the fact that WordNet does not conform to Li and Abe's notion of a thesaurus. To deal with 1, Li and Abe treat each noun sense as a virtual noun and use equation 2.9 to estimate the frequency of a class; that is, the count for a noun is split equally among the noun's senses, and the count for a class is the sum of the counts for the senses in that class. Estimates for the probability of a class are obtained using relative frequencies, as in 2.10.

Li and Abe's solution to 2 is to remove certain parts of the hierarchy, namely those parts that are dominated by senses whose synsets contain a noun in the data. For example, if *food* appeared in the data for the object position of *eat*, that part of the hierarchy dominated by $\langle\mathtt{food}\rangle$ would be removed. (But note this would only apply to the instance of WordNet corresponding to the object position of *eat*.) That way, each noun in the data appears in synsets that are at the leaves. Li and Abe's solution to 3 is to turn the DAG into a tree, by copying each subgraph with multiple parents. However, Li and Abe are not very clear about how the count for a noun should be distributed according to this modified structure. Note that problems 1 and 3 also affect the estimation method described in Chapter 3, but not problem 2, since we do not require the hierarchy to be in the form of a thesaurus.

Li and Abe apply their MDL techniques to structural disambiguation, which is described in Section 2.2.1, and also automatic clustering, which is described in Section 2.1.2. A further application is to investigate the validity of the usual assumption that arguments fill slots independently of the arguments in other slots (Li and Abe 1999).

Further applications of MDL

McCarthy adopts the methods of Li and Abe, primarily for the acquisition of diathesis alternations (McCarthy 1997; McCarthy and Korhonen 1998; McCarthy 2000). The basic approach to probability estimation is followed, although McCarthy does suggest some modifications that are worth

[6] Note that one reason for requiring the hierarchy to be a tree is that equation 2.12 does not necessarily hold for cuts in a DAG, and 2.12 is required for a sound application of MDL.

considering. The first modification is based on the following observation: that removing parts of the hierarchy based on the nouns that occur in the data can result in large parts being excised. For example, if *entity* appeared in the data, a large proportion of the complete hierarchy would be removed, namely that part of the hierarchy dominated by $\langle\texttt{entity}\rangle$. McCarthy's alternative solution is to create new leaf nodes for each internal node in the hierarchy; for example, the synset for the concept $\langle\texttt{entity}\rangle$ would be represented at a new leaf node having the internal $\langle\texttt{entity}\rangle$ node as a parent. This modification results in all the nouns in the hierarchy being represented at leaf nodes. Counts for nouns are distributed initially at leaf nodes and then 'passed up' to internal nodes representing the classes.

McCarthy's response to the DAG problem is to leave the hierarchy as a DAG and argue that, since only around 1% of the nodes in WordNet have more than one parent, the resulting tree cut models are unlikely to differ much from the tree case. McCarthy also notes that the majority of cases of multiple inheritance occur low down in the hierarchy, which have less effect on the resulting cut.[7]

Wagner (2000) also adopts the MDL approach for acquiring selectional preferences. One of his observations is that the level of abstraction of the cut depends, to a large extent, on the size of the data sample, so that large samples result in cuts near the leaves, and small samples result in cuts near the root. This behaviour is explained by the fact that, if a large amount of data has to be described, then the data description length tends to dominate, and cuts near the leaves have shorter data description lengths. Alternatively, if a small amount of data has to be described, then the model description length tends to dominate, and cuts near the root have shorter model description lengths. Wagner argues that this behaviour is undesirable, since the acquired preferences should not depend on the sample size. In order to counter this, Wagner modifies the encoding scheme so that the resulting cut is more robust to changes in sample size.

The generalisation procedure described in Chapter 3 is also affected by the sample size. However, we will argue that this is a positive feature of the procedure, since the point of generalisation in this thesis is not to acquire selectional preferences, but to estimate probabilities. As will be shown, the procedure is able to find those areas of WordNet where there are enough counts to give reasonable probability estimates. This issue will be discussed further in Chapter 3.

Encoding WordNet as a HMM

Abney and Light (1999) build on previous work in a couple of interesting respects: they are explicit about the underlying probability distribution generating the data, which allows them to estimate probabilities in a principled manner, and they attempt to carry out word sense disambiguation as a side effect of the estimation process. Abney and Light are critical of the work by Resnik, and Li and Abe, in that neither are very clear about the interpretation of the probability of a class. In particular, neither give a stochastic model for how the data are generated, and, without such a model, it is unclear how well (or by what method) the probabilities are being estimated.

Abney and Light's approach consists of encoding the WordNet hierarchy as a hidden Markov model (HMM), whose parameters can be estimated using well known techniques. (See Manning and Schütze 1999 for an introduction to HMMs.) The hierarchy is encoded as a HMM in a straightforward manner, by associating a state in the HMM with a node in the hierarchy, and a transition in the HMM with an arc in the hierarchy. Nouns in the data are then generated as follows: a "run" of the HMM begins at the root node, and child nodes are repeatedly chosen until a leaf node is reached, at which point a noun is generated from the leaf node. (In order that each noun in WordNet can be generated, Abney and Light create leaf nodes for each internal node, much like McCarthy.) Thus, there are two types of probabilities: transition probabilities that govern transitions between states, and emission probabilities that govern the generation of nouns from leaf nodes. A separate HMM is created for each verb and argument position, and each HMM generates the nouns for the corresponding verb and argument position in the data; the structure for

[7] Personal communication.

each HMM remains the same, but the values of the probabilities vary.

To give an example, consider how the noun *roll* is generated for the object position of *eat*. In fact, since *roll* has more than one sense in WordNet, there are numerous paths through WordNet that generate the noun, but let us assume that the noun is generated via the food sense. The hypernyms of the food sense of *roll* are as follows: $\langle$bread$\rangle$, $\langle$baked_good$\rangle$, $\langle$foodstuff$\rangle$, $\langle$food$\rangle$, $\langle$substance$\rangle$, $\langle$object$\rangle$, $\langle$entity$\rangle$. First, a child of the root of the hierarchy is chosen, in this case the $\langle$entity$\rangle$ node, according to the transition probabilities associated with the root. Then, the concept $\langle$object$\rangle$ is chosen, according to the transition probabilities associated with $\langle$entity$\rangle$. This process continues until a leaf node is reached, in this case $\langle$roll$\rangle$, at which point the noun *roll* is generated according to the emission probabilities associated with $\langle$roll$\rangle$. Note that *roll* is not the only noun that can be generated at this point, since emission probabilities are given to each noun in the synset of $\langle$roll$\rangle$: $\{$*bun*, *roll*$\}$. The probability of generating *roll* in this way is the product of the probabilities of the transitions multiplied by the probability of generating *roll* from the leaf node. The probability $p(roll|eat, \texttt{obj})$ (allowing any possible path to *roll*) is the sum of the probabilities of all ways of generating *roll*.

Abney and Light test this model on a word sense disambiguation task. Word sense disambiguation can be carried out using the HMM, since it is possible to calculate $p_{v,r}(s|n)$: the probability that sense s was used to generate noun n (relative to a verb, v, and argument position, r). To decide between the possible senses of n, the sense with the highest probability can be chosen. The same test and training data described in Resnik 1997 were used. The results were disappointing, however, since the model failed to outperform that of Resnik.

Abney and Light also use the HMM to model selectional preferences (based on the model of Resnik) and use equation 2.6 to measure the extent to which a class satisfies the preferences of a verb. The probability of a class is associated with the probability of being in the state in the HMM corresponding to that class. Abney and Light argue that the probability of being in a particular state at time t converges to a single value as t approaches ∞ (if additional transitions from each leaf state to the root state are added); it is this value that is taken to be the probability of the corresponding class. The distribution $p(c|v)$ can be obtained by training the HMM using nouns appearing with the verb (assuming some argument position), and $p(c)$ can be obtained by training the HMM using all the nouns in the data. Some examples of the acquired preferences are given, but, aside from the word sense disambiguation task, no formal evaluation is provided.

Much of Abney and Light 1999 is taken up with the fact that a straightforward application of the HMM training algorithm does not have the desired effect. In particular, it was hoped that using the Expectation Maximisation (EM) algorithm to estimate the parameters would achieve word sense disambiguation as a side-effect of the estimation, and that paths corresponding to the correct sense of a noun in the data would be favoured over paths corresponding to incorrect senses. What Abney and Light found is that, if the initial parameter settings account for a noun as a mixture of senses (using a uniform distribution across senses, for example), then there is no pressure to converge on parameter settings that favour a particular sense.

A further problem arises from the way nouns are assumed to be generated by a “run” through the HMM, since this leads to short paths being preferred over longer paths (because the probability of a path is a product of the probabilities of the transitions). Abney and Light respond to both problems by modifying the EM algorithm in various ways. The fact that various modifications are required leads Abney and Light to conclude that perhaps the EM algorithm is not the best choice for this application, despite the fact that it has become the default for uncovering hidden structure in natural language data.

Encoding WordNet as a Bayesian Network

Ciaramita and Johnson (2000), motivated by many of the same considerations as Abney and Light, attempt to acquire selectional preferences by encoding WordNet as a Bayesian network (Pearl 1988), rather than as a HMM. A separate network is encoded for each verb (and argument position). Each synset and word in WordNet is a node in the network, and each node represents a

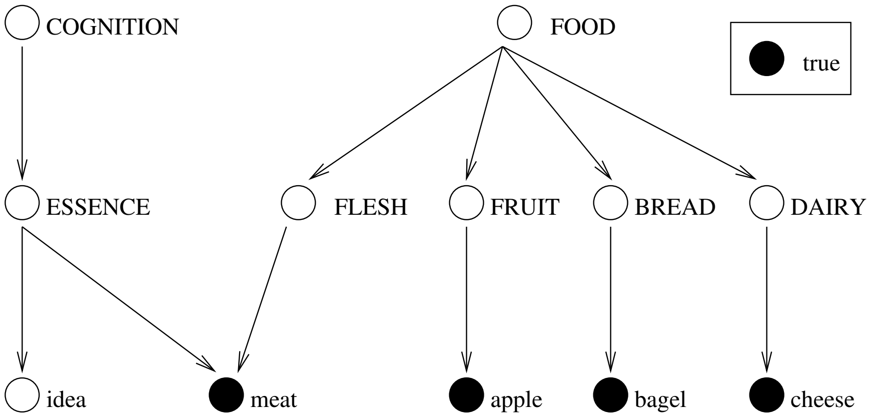

Figure 2.2: Example Bayesian network

variable, which can be in one of two states, *true* or *false*. A synset node has the value *true* if the concept represented by the synset is selected for by the verb, and a word node has the value *true* if the word can appear as an argument of the verb.

Each variable A, with parents $B_1,\ldots,B_n$, has associated with it a *conditional probability table* (CPT), which stores the probabilities $p(A|B1,\ldots,B_n)$. Ciaramita and Johnson call these probabilities the *priors*, and they are defined according to the following principles. First, it is *unlikely* that a verb selects for a concept, *a priori*. Second, if a verb does select for a concept, it is also *likely* that it selects for the hyponyms of that concept. *likely* and *unlikely* are given values that sum to one, such as 0.99 and 0.01 respectively. Similar principles apply to words: it is *likely* that a word can appear as an argument of the verb if the verb selects for any of the concepts whose synsets contain the word (i.e. the word's senses); and if the verb does not select for a concept, it is *unlikely* that the words in the concept's synset can appear as arguments of the verb. The values of the CPTs are set before any inference takes place.

Data consisting of words that occurred with the verb are used to "initialise" the network, so that the variables corresponding to words that appear in the data are set to *true*. Then, standard algorithms for training Bayesian networks can be used to infer posterior probabilities, using Bayes rule. The interesting thing about Bayesian inference in this context is that word sense disambiguation takes place as a side effect of the inference, through a property of Bayesian networks called "explaining away".

The following example is used by Ciaramita and Johnson. Suppose the network shown in Figure 2.2 represents the object position of the verb *eat*, and the following nouns have occurred with *eat* in the data: {*meat, apple, bagel, cheese*}. The variables corresponding to these nouns are set to *true*. With *likely* and *unlikely* set to 0.99 and 0.01 respectively, Ciaramita and Johnson calculate the posterior probabilities of FOOD and COGNITION to be 0.9899 and 0.0101 respectively. The example is designed to show that the FOOD node receives a high posterior probability, and the COGNITION node receives a low posterior probability. Thus the key point is that *meat* has provided evidence for the FOOD node but not the COGNITION node, so the correct sense of *meat* has been inferred. As Ciaramita and Johnson put it, the evidence has caused the "COGNITION hypothesis" to be *explained away*.

Part of the paper is concerned with computational issues, since the inference algorithms are expensive, and the CPTs require a lot of storage space. These problems are dealt with by only using a subpart of WordNet for each verb argument pair. The model was tested using the same word sense disambiguation task as Resnik (1997) and Abney and Light (1999), with favourable results, since the model outperformed both of these approaches.

2.1.2 Distributional similarity

The use of distributional similarity is an important alternative to using a man-made hierarchy for generalisation. The relevant literature is large, and we will only describe some representative approaches. Chapter 14 of Manning and Schütze 1999 also gives an overview of this area. After describing a number of approaches, we will consider the advantages and disadvantages of using distributional similarity, compared with using a man-made hierarchy for generalisation.

The philosophy underlying distributional approaches is that the probability of a rare event can be estimated by considering "similar" events that have occurred in the data. An example given by Lee and Pereira (1999) is that it is possible to infer that the bigram "after ACL-99" is plausible, even if it does not occur in the data, if "after ACL-95" does occur in the data. This assumes that "ACL-99" and "ACL-95" have similar cooccurrence distributions, or, in other words, that "ACL-99" and "ACL-95" tend to occur in the same contexts.

Similar events are often organised into clusters, according to some probabilistic measure of similarity. However, as Lee and Pereira (1999) point out, distributional approaches do not have to explicitly create clusters. Dagan, Lee, and Pereira (1999) estimate "cooccurrence probabilities" by taking the nearest cooccurrences to the target cooccurrence and averaging their probabilities. The cooccurrence can be between the head words in a syntactic construction, or between words in an *n*-gram, for example. Lee and Pereira (1999) call this approach *nearest-neighbors averaging*.

Following Dagan et al. (1999), let $W(w_1, w'_1)$ be a measure of the similarity between words w_1 and w'_1, and let $\mathcal{S}(w_1)$ be the set of words most similar to w_1; then $p(w_2|w_1)$ can be estimated as follows:

$$\hat{p}(w_2|w_1) = \frac{\sum_{w'_1 \in \mathcal{S}(w_1)} W(w_1, w'_1) p(w_2|w'_1)}{\sum_{w'_1 \in \mathcal{S}(w_1)} W(w_1, w'_1)} \tag{2.14}$$

The numerator is the probability of w_2 given a nearest neighbour of w_1 (weighted by a function of the similarity between w_1 and the neighbour) summed over all the nearest neighbours; and the denominator is a normalising constant.

There are a number of similarity measures, so rather than attempt to describe them all, we use one measure based on the Kullback-Leibler (KL) divergence as an example.[8] To measure the *dis*-similarity between two words, w_1 and w'_1, the KL divergence can be applied as follows:

$$\mathrm{D}(w_1 \| w'_1) = \sum_{w_2} p(w_2|w_1) \log \frac{p(w_2|w_1)}{p(w_2|w'_1)} \tag{2.15}$$

$\mathrm{D}(w_1 \| w'_1)$ can be transformed into a measure of similarity by using an appropriate function; Dagan et al. (1999), for example, apply the following transformation (where β is some constant):

$$W_{\mathrm{D}}(w_1, w'_1) = 10^{-\beta \mathrm{D}(w_1 \| w'_1)} \tag{2.16}$$

To make this clear, consider using 2.15 and 2.16 to compute the similarity of "ACL-99" and "ACL-95", assuming the adjective noun relationship. First, the KL-divergence compares $p(w_2|\text{ACL-99})$ and $p(w_2|\text{ACL-95})$ over all adjectives w_2. If the values of $p(w_2|\text{ACL-99})$ and $p(w_2|\text{ACL-95})$ are not very different across the different adjectives, then $\mathrm{D}(\text{ACL-99}\|\text{ACL-95})$ will have a small value. Then, the transformation in 2.16 takes this small dis-similarity value and produces a large similarity value.

Nearest neighbors averaging can be thought of as clustering taken to an extreme, in that each word in effect forms its own cluster (Lee and Pereira 1999). However, creating a cluster for each word means that the storage requirements for nearest neighbors averaging are typically quite high. In order to reduce the storage requirements, words can be generalised into a smaller number of representative clusters. There have been many suggestions for how this can be done, some of which are described below.

[8] Lee (1999) describes a number of possible similarity measures and compares their performance.

Clustering

Pereira, Tishby, and Lee (1993) acquire clusters of nouns for the direct object position of verbs. The clustering is “soft”, in that each word belongs to a cluster according to a cluster membership probability, and it is also “hierarchical”, in that the clustering algorithm works in a top-down, iterative fashion, splitting existing clusters at each iteration. The decision to keep two nouns in the same cluster is based on the difference between their conditional verb distributions, $p_n(v)$, which is measured using the KL divergence.

In contrast, Brown, Della Pietra, deSouza, Lai, and Mercer (1992) adopt a bottom-up iterative approach, in which initially the clusters are the individual words themselves, and the decision to merge two classes is based on the minimal loss of mutual information. The clustering is “hard”, in that a noun either belongs to a cluster or it does not, and there is no notion of degrees of membership. The clustering model was used to try and improve a language model, although no improvements in perplexity were gained by using a cluster-based as opposed to a word-based model. However, some improvement was obtained by using a linear interpolation between the word-based and cluster-based models.

Rooth, Riezler, Prescher, Carroll, and Beil (1999) use a similar clustering model to Pereira et al. (1993), but use the EM algorithm for estimation. Thus, in this framework, the problem is viewed as discovering hidden structure, where the observed, incomplete data is a set of verb-noun pairs, and the unobserved, complete data is a set of verb-noun-class triples. Rooth et al. argue that an advantage of using the EM framework is that it is mathematically well-defined and understood, whereas some of the other distributional approaches have not yet been given a clear probabilistic interpretation. The EM-derived clusters have been applied to the problem of word sense disambiguation (Prescher, Riezler, and Rooth 2000) and also statistical parsing using unification-based grammar formalisms (Riezler, Prescher, Kuhn, and Johnson 2000). Both papers present promising results. One of the results from Prescher et al. 2000 is that hybrid models combining frequency information with class-based information outperformed both pure frequency-based models and pure clustering models; this result accords to some extent with the results of Brown et al. (1992).

Li and Abe (1996) use the MDL Principle to cluster both nouns and verbs, given some data consisting of noun-verb pairs. Li and Abe’s model of the data assumes that the probability of a noun-verb pair is as follows:

$$p(n,v) = \frac{p(C_n, C_v)}{|C_n \times C_v|} \quad \text{for all } \; n \in C_n, v \in C_v \tag{2.17}$$

where C_n is the cluster containing n, C_v is the cluster containing v, and $|C_n \times C_v|$ is the cardinality of the cartesian product of C_n and C_v. Thus each noun-verb pair that can be created from the words in C_n and C_v is assumed to be equally likely. There is a trade-off between a model with a lot of clusters, which is likely to fit the data well, and a simpler model with less clusters, which is likely to fit the data less well. As was explained in Section 2.1.1, MDL is designed to handle exactly this kind of trade-off, and can be used to select an appropriate model.

Li and Abe evaluate the selected models using a PP-attachment task. In addition, the task performance using cluster-based estimates is compared with the performance using WordNet-based estimates (obtained using the method described in Section 2.1.1). It was found that the WordNet based estimates lead to a ‘confident’ attachment in more cases (where the level of confidence is a function of the difference between the probabilities assigned to the possible attachment sites), but that the cluster-based estimates lead to greater accuracy on those cases where a confident decision can be made. It was also found that, if the system is forced to make a decision on all cases (using a default decision if necessary), a combination of the two approaches slightly out-performs just using WordNet.

Summary

Two general methods of using distributional similarity have been described: nearest-neighbors averaging and clustering. Regarding the question of which provides better performance, Lee and

Pereira (1999) found that neither performed significantly better than the other on a simple decision task. (A modified version of the approach in Pereira et al. 1993 was used for the clustering, and various similarity measures were used for the nearest-neighbors averaging.) As Lee and Pereira comment, this result is in contrast to the widely held view that clustering is likely to perform less well than nearest-neighbors averaging (see, for example, Dagan, Marcus, and Markovitch 1995).

A more important question for this thesis is the comparison between distributional approaches and the use of a man-made hierarchy. It is an empirical question whether cooccurrence probabilities estimated using distributional methods, or probabilities of lexical sense preferences estimated using WordNet, provide better performance on some task; this is a question that has not been investigated and is an area for future research. The only work we are aware of that has addressed a similar question is that of Li and Abe (1996) (described above). However, it is still possible to offer some remarks contrasting the two approaches; the discussion below is based largely on that given in Resnik 1993a and Resnik 1998.

Resnik makes a number of observations regarding classes that are derived using distributional methods. He first notes that, unlike classes in WordNet, classes that are derived automatically are not associated with symbolic labels of any kind. This is not an issue if the classes are to be used only for probability estimation, but it is an issue if the classes are to be used for semantic interpretation. A related problem is that it is difficult to give a semantics for the derived classes; the words in a particular class often appear to be related, but it is not always clear exactly how they are related. According to Resnik (1998), the best way to describe the relationship between automatically clustered words is that they are “words that tend to appear in similar contexts,” but this is “no more than a restatement of the method by which the classes were derived.” This problem does not arise in the context of probability estimation, however.

An issue that is relevant to the estimation problem is that distributional methods do not always distinguish between the different senses of a word. The ‘hard’ clusters derived by Brown et al. (1992) have this property, in that a word can belong to only one cluster. Resnik argues that it is difficult to see how pairs like *school* and *grade* could be classified together, and also *grade* and *slope*, without putting words together that do not belong in the same class. The ‘soft’ clustering approaches, such as that of Pereira et al. (1993), are an attempt to overcome this problem. Another issue relevant to the estimation problem is that WordNet has been designed to be domain independent; in contrast, clusters derived automatically will reflect the characteristics of the training corpus, and will have to be re-acquired for each domain in which they are to be used. Domain dependence can lead to improved performance, of course, but note that probabilities estimated using WordNet will also reflect the training data. The advantage in using WordNet is that the same hierarchy can be used for each domain.

In conclusion, there are disadvantages and advantages to both approaches; however, for the purposes of probability estimation, the key question is really an empirical one, which has yet to be addressed.

2.1.3 Cooccurrence extraction

The term ‘cooccurrence’ is used here in a general sense, to mean two or more words that are likely to occur together in some context. Examples of possible contexts include syntactic contexts, such as a verb object relation, or simply consecutive words in a text. The reason cooccurrence extraction is relevant for this thesis is that the scores used for measuring the strength of a cooccurrence could also be used to measure the strength of association between a noun sense and a predicate. (Although note that reasons were given in Chapter 1 for preferring probabilities over such scores.) Also, later chapters use statistics that are related to some of those described below, all of which have been used in the context of cooccurrence extraction.

Mutual Information

The mutual information between two words x and y (in some cooccurrence relation) is defined as follows:

$$I(x,y) = \log_2 \frac{p(x,y)}{p(x)p(y)} \tag{2.18}$$

The mutual information described here is often referred to as *pointwise mutual information*, to distinguish it from the notion used in information theory. Pointwise mutual information is derived from the information-theoretic notion, but the information-theoretic version is defined as an average over random variables. Also, the pointwise version has less of a theoretical basis; Jelinek (1997) warns that interpreting $I(x,y)$ as the mutual information between x and y gives "only an intuitive interpretation." (p.134)

Pointwise mutual information compares the joint probability of observing x and y together, $p(x,y)$, with the probability of this observation if x and y were independent, $p(x)p(y)$. If x and y are highly associated, we would expect $p(x,y)$ to be much greater than $p(x)p(y)$, resulting in a large positive value for $I(x,y)$. If x and y are not related, we would expect $p(x,y) \approx p(x)p(y)$ and $I(x,y) \approx 0$. And if x and y are 'negatively associated', we would expect $p(x,y)$ to be less then $p(x)p(y)$ and $I(x,y)$ to be less than zero. (Although note that, in practice, it is difficult to arrive at an *estimated* value that is less than zero, unless the training set is very large (Dagan et al. 1995).)

The use of pointwise mutual information as a measure of cooccurrence was advocated in a number of early papers. Hindle (1990) suggested using it to group nouns into semantic classes, on the basis of their mutual information scores averaged over different verbs and argument positions; so *boat* and *ship* might be grouped together, for example, because they are likely to have similar mutual information scores relative to the subject position of *cruise*, and the object position of *sink*, and so on. Church and Hanks (1990) use a variant of mutual information, which they call the 'association ratio', to find associated pairs of words that occur together in some fixed window size. Their approach can find nouns related semantically, such as *doctor* and *nurse*, but can also find words related syntactically, such as phrasal verbs like *set up, set off, set out*, and so on. In more recent work, Lin (1998) uses mutual information as a measure of similarity for semantic clustering, and Dagan et al. (1995) use mutual information in a nearest neighbor approach to cooccurrence probability estimation.

Magerman and Marcus (1990) suggested using mutual information scores (based on part of speech tags) for parsing. However, in later work Magerman used a probability model instead (Magerman 1994, 1995). One advantage of using a probability model is that the rules of probability theory can be used to combine probabilities, whereas there is no established theory of how to combine mutual information scores. This point is important when defining a statistical model for complex linguistic events such as parse trees. Another advantage is that additional information, such as structural information relating to the parse, can be integrated into a probability model in a theoretically sound way. Similar reasons were given in Chapter 1 for representing lexical sense preferences using probabilities, rather than mutual information or related scores.

There is another potential problem with using mutual information scores, which is that estimates based directly on counts can be subject to over-estimation when the counts are small (Dunning 1993; Manning and Schütze 1999, p.181). Since natural language data invariably leads to some small counts, this is an important issue. A score very similar to mutual information is used in Chapter 4 as a measure of association between verbs and WordNet classes. However, we will argue that, on the whole, the measure is only applied to classes with enough counts to lead to a reliable estimate.

Contingency table statistics

Dunning (1993) considers how statistics defined over a 2×2 contingency table can be used to identify highly associated bigrams. (Dunning's technique also applies to other types of cooccurrence relation.) Table 2.1 shows a contingency table corresponding to the bigram $w_1 w_2$, where

$f(w_1, w_2)$	$f(\neg w_1, w_2)$
$f(w_1, \neg w_2)$	$f(\neg w_1, \neg w_2)$

Table 2.1: Contingency table for the bigram $w_1 w_2$

$f(w_1, w_2)$ is the number of times w_2 follows w_1 in the data, and $f(\neg w_1, w_2)$ is the number of times w_2 follows a word other than w_1 in the data. (The other frequencies in the table are defined analogously.) The null hypothesis corresponding to the table is that w_1 and w_2 appear independently of each other, and a statistic such as chi-squared can be used to determine how likely the null hypothesis is to be true. If the chi-squared statistic has a high value, then this gives strong evidence that the null hypothesis is false, and that w_1 and w_2 are highly associated. Thus bigrams with high chi-squared scores should correspond to highly associated pairs of words or collocations.

The chi-squared statistic that is usually encountered in text books is the *Pearson* chi-squared statistic. However, the problem with this statistic, as Dunning demonstrates, is that it can over-estimate the significance of rare events. This means that the bigrams producing the highest scores are often based on very low counts, which makes the test unreliable. Most of the top ranked bigrams in Dunning's experiments occurred only once in the data, and among the highest ranked bigrams were cases like *practically drawn*, *instance 280* and *scanner cash*, which are hardly highly associated pairs of words. As a remedy to this problem, Dunning considers the log-likelihood ratio statistic, denoted G^2, which does not over-estimate the significance of rare events in the same way. The top ranking bigrams produced according to this statistic were much more intuitive.

Dunning's analysis of his results is based on the following claim: that the sampling distribution of G^2 approaches chi-squared quicker than the sampling distribution of X^2. However, this part of Dunning's analysis is debatable, since Agresti (1996) makes exactly the opposite claim:

> The sampling distributions of X^2 and G^2 get closer to chi-squared as the sample size n increases ... The convergence is quicker for X^2 than G^2. (p.34)

Given Aresti's comments, a more likely explanation lies in the conservative nature of G^2, which means that X^2 is more likely to return a significant result for a table based on small counts. This would explain Dunning's results, in which pairs of words occurring infrequently in the corpus obtain high scores according to X^2 but not G^2. These issues will be discussed further in Chapter 3, where a chi-squared test is used as part of a procedure for selecting a suitable level of abstraction in WordNet.

Pedersen (1996) suggests using Fisher's exact test (Agresti 1996) for bigram discovery, rather than a chi-squared statistic. The advantage of Fisher's exact test is that it can be applied to any contingency table, regardless of the size of the counts, and the result will be reliable. However, the test is computationally expensive, since it involves computing every contingency table that could have led to the marginal totals observed in the sampled table. (The marginal totals are not shown in Table 2.1, but are simply the totals obtained by summing the scores in each row and column.) In addition, the results obtained by Pedersen for the exact test did not differ greatly from those obtained for the log-likelihood statistic, and so it is not clear that the benefits of using the test outweigh the additional computational burden.

2.1.4 Smoothing for probability estimation

Many of the smoothing techniques used in corpus-based NLP were developed for language modelling, and so to demonstrate some of the most widely used techniques, we consider the problem of estimating an n-gram model. More specifically, the problem is to estimate the probability of a word conditional on the previous $n-1$ words: $p(w_i | w_{i-n+1} \dots w_{i-1})$. A maximum likelihood estimate of this probability is as follows:

$$\hat{p}(w_i|w_{i-n+1}\dots w_{i-1}) = \frac{f(w_{i-n+1}\dots w_i)}{f(w_{i-n+1}\dots w_{i-1})} \tag{2.19}$$

where $f(w_{i-n+1}\dots w_i)$ is the frequency of the n-gram $w_{i-n+1}\dots w_i$ in the data. Clearly, this approach will result in many zero probability estimates, even for a large data set. The task of smoothing is to reduce the problem of over-fitting, by giving some of the probability mass assigned to seen cases to unseen cases.

Additive smoothing

Additive smoothing is the simplest type of smoothing used in practice, where, in order to avoid zero probability estimates, a constant δ is added to the count for each n-gram ($|V|$ is the size of the vocabulary):

$$p_{\text{add}}(w_i|w_{i-n+1}\dots w_{i-1}) = \frac{f(w_{i-n+1}\dots w_i)+\delta}{f(w_{i-n+1}\dots w_{i-1})+\delta\,|V|} \tag{2.20}$$

The term $\delta\,|V|$ in the denominator is a normalising factor to ensure that the probability estimates still sum to one. This approach has a long history; indeed, Manning and Schütze (1999) attribute such an approach to Laplace and cite a text from 1814. Laplace suggested a value of one for δ, in which case the method is often referred to as *adding one*. Using a value of one has the problem that too much probability mass tends to be given to unseen cases, especially for very large vocabularies. A more popular value for δ is $1/2$ (Manning and Schütze, 1999; p.204). However, Gale and Church (1990, 1994) argue that additive smoothing applied to language data performs poorly, whatever the value of δ.

One of the problems with additive smoothing is that the same probability estimate is assigned to all unseen events (relative to some probability distribution). As an example, consider using additive smoothing to estimate the probabilities of lexical sense preferences, for the subject position of *run*. If $\langle$`fox`$\rangle$ and $\langle$`carpet`$\rangle$, say, do not appear in the subject position of *run* in the data, then additive smoothing will give these two senses the same probability estimate. But we would expect $\langle$`fox`$\rangle$ to have a higher probability than $\langle$`carpet`$\rangle$ for this verb and argument position. The estimation method developed in Chapter 3 is an attempt to distribute probability mass in a more motivated fashion, using classes from WordNet to try and determine those unseen senses that are likely to occur with the verb.

Deleted interpolation

A more widely used smoothing technique is a form of linear interpolation, in which a maximum likelihood estimate based on a history of $n-1$ words is combined with estimates based on less history:

$$p_{\text{interp}}(w_i|w_{i-n+1}\dots w_{i-1}) = \lambda_{w_{i-n+1}^{i-1}}\hat{p}(w_i|w_{i-n+1}\dots w_{i-1}) + (1-\lambda_{w_{i-n+1}^{i-1}})p_{\text{interp}}(w_i|w_{i-n+2}\dots w_{i-1}) \tag{2.21}$$

This formulation is taken from Chen and Goodman 1996. Note that the estimate of the probability $p(w_i|w_{i-n+2}\dots w_{i-1})$, which has one less word of history, has also been smoothed in the same way. The intuition behind the approach is that the values of the λ's will reflect the reliability of each estimate. This method of linear smoothing is often called *deleted interpolation*. Jelinek and Mercer 1980 and Bahl, Jelinek, and Mercer 1983 are early papers describing this technique, and they give ways in which the values of the λ's can be estimated. A more recent text book treatment is given in Jelinek 1997.

Linear interpolation could be used to estimate the probabilities of lexical sense preferences, $p(c|v,r)$, where c is a noun sense, v is a predicate and r is an argument position. One way to combine estimates based on less context is as follows:

$$p_{\text{interp}}(c|v,r) = \lambda_1\hat{p}(c|v,r) + \lambda_2\hat{p}(c|r) + \lambda_3\hat{p}(c) \tag{2.22}$$

As an example, consider using 2.22 to estimate $p(\langle\texttt{fox}\rangle|run,\text{subj})$ and $p(\langle\texttt{carpet}\rangle|run,\text{subj})$, assuming that neither $\langle\texttt{fox}\rangle$ nor $\langle\texttt{carpet}\rangle$ appear with *run* in the data. Unlike additive smoothing, the two unseen senses are unlikely to receive the same estimate, since the estimates based on less context are unlikely to be the same for the two senses. However, $\langle\texttt{fox}\rangle$ will not necessarily receive a higher estimate than $\langle\texttt{carpet}\rangle$; the problem is that the estimates based on less context ignore the verb. In contrast, the estimation method presented in Chapter 3 is able to make use of the verb, by determining whether semantically similar senses to $\langle\texttt{fox}\rangle$ and $\langle\texttt{carpet}\rangle$ appear as subjects of *run.*

Good-Turing

Another widely-used technique is the Good-Turing method (Good 1953), which states that an *n*-gram that has occurred *r* times in the data should have an adjusted frequency r^*, where

$$r^* = (r+1)\frac{E(N_{r+1})}{E(N_r)} \quad (r \geq 1) \tag{2.23}$$

$E(N_r)$ is the expected number of *n*-grams that occur *r* times in the data. Relative frequencies based on the r^* values can be used to estimate the probabilities. Note that 2.23 only applies to values of *r* greater than zero; a further result of Good 1953 is that the total probability mass assigned to unseen objects is $E(N_1)/N$, where N is the total number of *n*-grams in the data.

In practice, the actual number of *n*-grams that occur *r* times in the data, n_r, can be used to approximate the expected values, if the actual values are suitably smoothed themselves. To see that some smoothing is required, note that, for the most frequent *n*-gram in the data, n_{r+1} is zero. Substituting a value of zero for $E(N_{r+1})$ in 2.23 leads to a value of zero for r^*, and hence a zero probability estimate. Clearly, a zero probability estimate for the most frequent *n*-gram is not what is required. Gale and Sampson (1995) give a simple technique for smoothing the n_r; an appendix in Church and Gale 1991 gives the mathematics behind the approach.

Note that the Good-Turing method says nothing about how to divide the reserved probability mass among the unseen items. One simple approach would be to divide it equally, but there are more sophisticated approaches; Church, Gale, Hanks, and Hindle (1991) and Gale and Sampson (1995) discuss some possibilities. Chen and Goodman (1996) comment that Good-Turing is not usually used directly for *n*-gram smoothing, because good performance in language modelling is obtained by considering models of varying order (as in deleted interpolation). However, it is often used in combination with other techniques, and a widely used method that combines Good-Turing with variable order models is the backing-off technique of Katz (1987).

Katz backing-off

Katz does not combine different order models, as in deleted interpolation, but rather chooses between them. The choice is made by considering estimates based on progressively shorter histories, and the first reliable estimate encountered during the backing off is chosen. The reliability of an estimate is determined by considering the frequency of the *n*-gram in the training data; for example, if $p(w_i|w_{i-n+1}\dots w_{i-1})$ is being estimated, and the *n*-gram $(w_{i-1}\dots w_i)$ occurs many times in the data, then the estimate based on the history of $(n-1)$ words is likely to be reliable. If the *n*-gram occurs only a small number of times, then an estimate based on less history is considered. This process is repeated until a reliable estimate is found. Good-Turing is used to reserve some probability mass for the unseen *n*-grams whose probabilities are estimated by backing off.

Chen and Goodman (1996) comment that Katz smoothing "is perhaps the most widely used smoothing technique in speech recognition", and this technique did perform well in Chen and Goodman's experiments. More recently, Dagan et al. (1999) have shown that a distributional similarity-based approach outperforms Katz smoothing for bigram probability estimation, and also on a simple pseudo-disambiguation task.

2.2 Applications

This section describes previous work on structural disambiguation, which is a problem considered later in the thesis. The section describes work on PP-attachment, and then work that has considered the more general problem of parse selection. Not all previous approaches are considered, since the literature in both cases is very large, and we describe only those approaches that are most relevant to the work in this thesis.

2.2.1 Structural disambiguation: PP-attachment

The type of structural ambiguity that has been most covered in the literature is PP-attachment ambiguity. This is a pervasive form of ambiguity, and a potentially damaging one, in that increasing the number of PPs in a sentence can lead to a combinatorial explosion in the number of possible analyses (Church and Patil 1982). A number of early studies in the psycholinguistics domain suggested possible strategies for resolving attachment ambiguities. Two of the most cited studies are those of Kimball (1973), who suggested that a constituent tends to attach to another constituent immediately to its right (right association), and Frazier (1978), who suggested that there is a preference for attachments that lead to the parse tree with the fewest nodes (minimal attachment). However, later work (Whittemore, Ferrara, and Brunner 1990; Taraban and McClelland 1988) demonstrated that lexical information is a better predictor of attachments, and most of the recent corpus-based approaches to structural disambiguation, including PP-attachment, have been based on lexical associations.

The PP problem that is usually addressed only considers sequences of the following form: (*verb, direct object of verb, preposition, object of preposition*). Moreover, only the heads of the noun phrases are usually considered. The problem can then be characterised as as taking a four-tuple, (v, n_1, pr, n_2), and deciding whether the PP attaches to v or n_1, as in the much used example (*see, man, with, telescope*). Note that this is an easier problem than the most general form of PP-attachment, since only two possible attachment sites are being considered. In the general case, there may be more than two sites. Consider this example from Hindle and Rooth (1993):

(2.24) NBC was so afraid of hostile advocacy groups and unnerving advertisers that it shot its dramatization of the landmark court case that legalised abortion under two phony script titles.

The PP headed by *under* could attach to *abortion* or *legalised*, but in fact attaches to the higher verb *shot*. Franz 1996 is one of the few examples of work that considers the more general problem.

Hindle and Rooth 1991, 1993

The original corpus-based approach that motivated much of the later work on PP-attachment is that of Hindle and Rooth (1991, 1993). The basis of their approach is to compare $p(pr|v)$ and $p(pr|n_1)$; so to decide on the attachment point for (*send, soldiers, into, Afghanistan*), the probabilities $p(into|send)$ and $p(into|soldiers)$ are compared. Intuitively, the idea is to consider whether *send into* or *soldiers into* is a more likely combination. In Hindle and Rooth 1991, a t-statistic is used to compare the probabilities, so that the method can decline to make a decision if the difference in the two probabilities is not statistically significant. A similar approach is applied in Hindle and Rooth 1993, but using a log likelihood ratio rather than a t-score.

The probabilities are estimated using a partially supervised boot-strapping approach, where a partial parser is used to identify prepositional phrases attached to verbs or nouns in a corpus. Hindle and Rooth used the Fidditch parser (Hindle 1994), together with a corpus of Associated Press news stories. Some of the attachments will be unambiguous, and these cases can be used to build up counts that can be used to resolve (some of) the ambiguous cases. If the log likelihood ratio is not significant for an ambiguous case, equal credit is given to each attachment point. The performance reported in Hindle and Rooth 1993 is around 80% on unseen text, if the classifier is forced to make a decision on all test cases. This precision score can be increased (but at the expense of recall) by only making decisions when the log likelihood ratio (or t-score) exceeds a certain threshold.

One obvious problem with Hindle and Rooth's approach is that it completely ignores n_2, which can be harmful in some cases. Consider this example from Resnik 1993a:

(2.25) Britain reopened its embassy in December.

(2.26) Britain reopened its embassy in Teheran.

In 2.25, the attachment is to the verb, but, in 2.26, the attachment is to the noun. Just comparing $p(pr|n_1)$ and $p(pr|v)$ would result in the same attachment decision in both cases. The obvious extension is to compare $p(pr,n_2|n_1)$ and $p(pr,n_2|v)$; however, Resnik (1993a) comments that "Attempts to simply compare p$(p,n2|n1)$ against p$(p,n2|v)$ using the t-score fail dismally, and there is no reason to think the log likelihood ratio would fare any better." (p.116) But this comment assumes that Hindle and Rooth's estimation technique is to be used to estimate the probabilities. In Chapter 6, we extend Hindle and Rooth's approach by incorporating n_2 in the obvious way, and use the class-based method from Chapter 3 to estimate the probabilities. Presumably, Hindle and Rooth chose to ignore n_2 because of the sparse data problem caused by introducing it, but our estimation method has been designed to deal with this kind of problem.

Collins and Brooks 1995

Collins and Brooks (1995) estimate probabilities of the form $p(A|v,n_1,pr,n_2)$, where A is the attachment site (v or n_1), and choose the site corresponding to the highest probability. A backing-off approach, motivated by the work of Katz (1987) for language modelling, is used to estimate the probabilities. The idea is that, if the test tuple (v,n_1,pr,n_2) has appeared at least k times in the training data, then the estimate is based directly on that tuple:

if $f(v,n_1,pr,n_2) > k$

$$\hat{p}(A|v,n_1,pr,n_2) = \frac{f(A,v,n_1,pr,n_2)}{f(v,n_1,pr,n_2)}$$

If (v,n_1,pr,n_2) has not appeared more than k times, then the classifier uses an estimate based on less context, where initially the backed-off context consists of three words. However, there are a number of contexts involving three words, and so a choice of context has to be made. What Collins and Brooks found (perhaps not surprisingly) is that the preposition is crucial for making the attachment position, and so frequencies based on those three word contexts including the preposition were used:

else if $f(v,n_1,pr) + f(v,pr,n_2) + f(n_1,pr,n_2) > k$

$$\hat{p}(A|v,n_1,pr,n_2) = \frac{f(A,v,n_1,pr) + f(A,v,pr,n_2) + f(A,n_1,pr,n_2)}{f(v,n_1,pr) + f(v,pr,n_2) + f(n_1,pr,n_2)}$$

The approach then backs off to a two word context, followed by a one word context, and finally a default decision:

else if $f(v,pr) + f(n_1,pr) + f(pr,n_2) > k$

$$\hat{p}(A|v,n_1,pr,n_2) = \frac{f(A,v,pr) + f(A,n_1,pr) + f(pr,n_2)}{f(v,pr) + f(n_1,pr) + f(pr,n_2)}$$

else if $f(pr) > k$

$$\hat{p}(A|v,n_1,pr,n_2) = \frac{f(A,pr)}{f(pr)}$$

else

$$\hat{p}(A|v,n_1,pr,n_2) = 1 \text{ if } A \text{ is noun attach, } 0 \text{ if } A \text{ is verb attach}$$

An interesting result of the paper is that the optimum value for k was found to be zero at all stages. This means that, even if a context occurs only once in the training data, it is better to use an estimate based on that context, rather than back off to another level. We present a related result in Chapter 6, regarding the use of low count events in the training data. We find that, for a simple pseudo-disambiguation task, it is better to use counts based on classes that are low down in the hierarchy (where the data are likely to be noisy and sparse), rather than 'back off' to counts based on classes that are high in the hierarchy (where the data are less sparse). Daelemans, Van Den Bosch, and Zavrel (1999) present a similar result regarding the use of low count training events, in the context of instance based learning.

The test and training data used by Collins and Brooks first appeared in Ratnaparkhi, Reynar, and Roukos 1994, and have now become the standard data for this task. The data were taken from the Penn Treebank (Marcus, Santorini, and Marcinkiewicz 1993), and consist of $20,801$ (v,n_1,pr,n_2,A) tuples for training (where A is the correct attachment site), and $3,097$ tuples for testing. A useful result reported by Ratnaparkhi et al. is that the human performance on a smaller but similar test was around 88%, if the subjects were given only the four head words, and 92% if the subjects were exposed to the whole sentence. These figures provide useful upper bounds for the task. Collins and Brooks reported an accuracy of 84.1%, which increased to 84.5% if various strategies were used to pre-process the data, such as lemmatizing the words, replacing strings of numbers with the token 'NUM', and replacing proper names with 'NAME'. These results presented a significant improvement over previous approaches.

Stetina and Nagao 1997

The motivation for the approach of Stetina and Nagao (1997) is that the accuracy of the Collins and Brooks method is extremely high for those cases where a four word or three word match is found in the training data (92.6% and 90.1%, respectively, when no pre-processing of the data is used). This suggests that if the number of three or four word matches could be increased, the performance would improve. Stetina and Nagao's suggestion is to increase the number of matches by including cases that are semantically close to the target case. The example they use is that *Buy books for children* should be matched with *Buy magazines for children*, since books and magazines are semantically similar.

This is how Stetina and Nagao motivate their approach, although in practice their approach is very different to that of Collins and Brooks. The training consists of inducing a number of decision trees (one for each preposition), which involves splitting the training examples into groups according to their semantic similarity. In order to compare the semantic similarity of examples, a word sense disambiguation algorithm is developed, together with a heuristic distance metric that measures the distance between two senses in a semantic hierarchy. The noun and verb hierarchies from WordNet were used to supply the senses.

The induced decision trees can then be used to classify unseen cases, by effectively looking at the attachments of semantically similar cases. The overall result was a success rate of 88.1%, using the standard training and test sets, which matches the human performance on this task. There is a disadvantage to Stetina and Nagao's approach, which is that the training phase (particularly the disambiguation of the training set) is computationally very expensive. However, the training only needs to be performed once, and, in order to speed up the word sense disambiguation of the test cases, a less expensive version of the algorithm was used for the test cases.

The performance of the system described in Chapter 6, which also uses WordNet to resolve PP-attachment ambiguities, is below that of Stetina and Nagao. However, there are a number of differences between the two approaches. First, Stetina and Nagao's system uses a complex disambiguation algorithm based on decision trees, whereas the system described in Chapter 6 simply compares probabilities corresponding to the possible attachment sites. An advantage of our approach is that these probabilities can be easily integrated into a model for parse selection (see Chapter 5 for an example of how this can be done). Stetina and Nagao's classifier could also be used as part of a parse selection system, but only if the parsing is done in stages, by first identifying constituents, and then in a later phase deciding where certain constituents attach. A further advantage of our approach is that an unsupervised training method can be used (such as that devised by Ratnaparkhi (1998)), which means that the system can potentially be applied to languages for which there are no treebanks. In contrast, Stetina and Nagao's system requires supervised training data.

Ratnaparkhi 1998

A paper that takes a slightly different perspective on the PP-attachment problem is Ratnaparkhi 1998. Ratnaparkhi shows that it is possible to get reasonable performance – 81.9% on the standard test set – by training on unsupervised data. This result shows that it may be possible to build fairly accurate classifiers for languages for which there are no treebanks. As well as evaluating on the standard test set, Ratnaparkhi also gives an evaluation for Spanish.

The unsupervised training is motivated by a step in the boot-strapping procedure of Hindle and Rooth, in that counts are obtained from unambiguous cases of attachment. However, unlike Hindle and Rooth, Ratnaparkhi uses the unambiguous cases only, and does not attempt to resolve ambiguous cases for the purposes of training. The unambiguous cases are identified by applying a simple heuristic to tagged and chunked text. An example given in the paper is for the following sentence (already reduced by a chunker): *conduct of lawyers in jurisdictions is guided by rules* … . Here, the extraction heuristic would return *lawyers in jurisdictions* as an example of noun attachment, and *guided by rules* as an example of verb attachment. The heuristic is able to identify these cases because there is no preceding verb in the noun attachment case, and no intervening noun in the verb attachment case, and hence there is no attachment ambiguity. The phrase *conduct of lawyers* would also be identified as a case of noun attachment, under the assumption that PPs headed by *of* always attach to the noun. Ratnaparkhi applied the heuristic to Treebank data to evaluate its accuracy, and found that only 69% of the attachments returned by the heuristic were correct (excluding those cases where the preposition is *of*). However, an advantage of using unsupervised data is that a large volume can be produced, which is able to offset the lower accuracy to some extent.

This completes the relevant literature on PP-attachment,[9] except for two approaches that use WordNet, and which are the closest to the approach presented in Chapter 6. These are from Li and Abe (1998) and Resnik (1993a) (see also Resnik 1993b and Resnik and Hearst 1993).

Resnik 1993a, 1993b, Resnik and Hearst 1993

The intuition behind Resnik's approach is similar to that of Stetina and Nagao (1997), in that words are grouped together into semantically similar classes, using the noun hierarchy of WordNet. After choosing semantic classes for n_1 and n_2, the attachment decision is made on the basis of a *conceptual association* measure, which is similar to the selectional association measure described in Section 2.1.1. The difficulty is in choosing the semantic classes, since nouns exhibit 'horizontal' and 'vertical' ambiguity, in the sense described earlier.

To explain Resnik's solution, consider the example *augment staff in Dallas*. All possible classes for *staff* and *Dallas* are considered, and the conceptual association is compared for each pair and the two attachment sites. So all classes for *staff* are considered (across all senses of *staff*): SOCIAL_GROUP, FACULTY, IMPLEMENT, BODY, MUSICAL_NOTATION … , and all classes for *Dallas*: URBAN_AREA, REGION, CITY … , and the attachment site is chosen that tends to score higher across the different pairs. This explanation is a little crude, and Resnik 1993a should be consulted for the exact details, but it expresses the underlying approach.

[9] Some of the work we have not included is the following: (Ratnaparkhi et al. 1994; Brill and Resnik 1994; Zavrel, Daelemans, and Veenstra 1997; Abney, Schapire, and Singer 1999; Alegre, Sopena, and Lloberas 1999).

This strategy performed less well than that of Hindle and Rooth on Penn Treebank data, but it was found that a combination of the two provided the best results. The combination was to use Hindle and Rooth's method first, and back off to Resnik's method if the t-score was not significant. Resnik presents a result of 83.9% for this combined method on a fairly small test set; however, to put these results in perspective, Brill and Resnik present an accuracy of only 76.0% on the Treebank data used in Brill and Resnik 1994.

The disambiguation method presented in Chapter 6 is similar to Resnik's, but probabilities are compared rather than association measures, and a class is chosen only for n_2 (and not n_1). We suspect that the additional noise introduced by considering classes for both nouns outweighs the benefit of using an additional class.

Li and Abe 1998, Li 1996

Li and Abe's (1998) approach (at least in principle) is to compare $p(n_2|v,pr)$ and $p(n_2|n_1,pr)$, and choose the attachment site (v or n_1) corresponding to the highest probability. In fact, since Li and Abe use WordNet to estimate the probabilities, they do not use probabilities of nouns, but use probabilities of classes instead.[10] MDL is used to obtain a tree cut model, as described in Section 2.1.1, and the class that contains n_2 is chosen from the classes in the cut. In fact, since n_2 may be ambiguous, and thus belong to many classes in the cut, the class is chosen that provides the highest probability estimate.

To make this clear, consider the example *protect house against damage*. MDL would be used to create tree cut models for the combinations *protect against* and *house against*, and a class containing *damage* would be chosen in each case. Since *damage* has a number of senses in WordNet, and thus could belong to a number of classes in each cut, the class is chosen that maximises the relevant probability in each case. So for the combination *protect against*, the class is chosen that maximises $p(C|protect,against)$, where C ranges over those classes that contain *damage* from the cut for *protect against*. Finally, the probabilities $p(C_v|protect,against)$ and $p(C_{n_1}|house,against)$ are compared, where C_v and C_{n_1} are the chosen classes corresponding to each attachment site.

The results were 82.2% accuracy using Penn Treebank data (but not the standard training and test sets). The score for a method similar to that of Hindle and Rooth was 80.7%, and so incorporating n_2 in this way resulted in only a modest improvement. Li (1996) reports better results, obtained by combining the lexical approach with a structural approach, where the structural approach uses attachment principles from the psycholinguistics literature (Kimball 1973; Hobbs and Bear 1990).

The disambiguation method described in Chapter 6 is similar to that of Li and Abe, except that our method compares joint probabilities of a noun sense and preposition, rather than probabilities conditioned on the preposition. To see why this difference is important, consider the example *eat slice of beef*. Li and Abe would compare $p(C_1|slice,of)$ and $p(C_2|eat,of)$ (where C_1 and C_2 are classes containing *beef*), and we would compare the probabilities $p(\langle\texttt{beef}\rangle,of|slice)$ and $p(\langle\texttt{beef}\rangle,of|eat)$. The problem with Li and Abe's approach is that the combination *eat of* is highly unlikely to occur in a corpus, and so there will be no indication in the training data of whether *beef* is a likely argument of *eat-of*. Moreover, by conditioning on the preposition, Li and Abe are ignoring the fact that a PP headed by *of* is highly unlikely to attach to the verb (especially when the verb is *eat*). Our method is able to use this information, and if a predicate-preposition combination occurs only rarely in the training data (in which case there will be little information about the possible arguments of that predicate-preposition combination), the method automatically 'backs off' to a probability that ignores n_2. This property of the approach will become clear in Chapter 6.

[10] Li and Abe use probabilities of classes, rather than probabilities of individual senses, since they claim that doing so gives a better result in practice.

2.2.2 Parse selection

The problem of parse selection is to select the correct parse for a sentence from a number of alternatives. As Collins (1999; p.6) points out, this can be an "astonishingly severe problem" in broad domains such as the Wall Street Journal (WSJ). Collins cites a number of factors that are responsible for the severity of the problem: the need for a large grammar to obtain broad coverage; long sentences being typical in a broad domain; and many common sources of syntactic ambiguity, such as PP-attachment, leading to exponential growth in the number of analyses (relative to sentence length). There are many examples in the literature of ordinary looking sentences having hundreds, sometimes thousands, of different analyses according to some grammar. The parser of Briscoe and Carroll, which is used in Chapter 5, produces 602 different analyses for the following sentence taken from the Susanne corpus (Sampson 1995): "He will be succeeded by Rob Ledford of Gainesville, who has been an assistant more than three years."

In response to this problem, NLP researchers began developing statistical methods in which a hand annotated corpus, or treebank, is used to estimate the parameters of a probabilistic parsing model. Under this approach, the correct parse is assumed to be the most probable parse according to the model. The first attempts to build statistical parsers were based on probabilistic versions of context free grammars (PCFGs), but the results were disappointing. Charniak (1997) reports precision and recall scores of around 74% for a PCFG trained and tested on the Penn Treebank (using the Parseval measures (Harrison et al. 1991)). It is now known that a major factor in the poor performance of these models is the lack of lexicalisation. We have already seen how lexical information is important for resolving ambiguities such as PP-attachment, and researchers have extended these ideas to the more general problem of parse selection; indeed, almost all of the most cited statistical parsers incorporate lexical dependencies in some form (Jelinek et al. 1994; Magerman 1995; Collins 1996, 1997; Eisner 1996b; Goodman 1997; Bod 1998; Ratnaparkhi 1999; Charniak 1997, 2000). The scores of 74% for a PCFG compare with scores in the mid to high 80s for the lexicalised models. Charniak (2000) has recently reported the first precision and recall scores over 90% using the standard training and test sets from the Penn Treebank.

We will now describe some approaches to statistical parsing, concentrating on those that have motivated the approach taken in Chapter 5. The parsing literature is vast, and it cannot all be covered here.[11] A more comprehensive review of the parsing literature can be found in Collins 1999.

Magerman 1995

As Collins (1999; p.108) points out, the work of Magerman (1995) represented a major advance in statistical parsing. Magerman's parser (which he called SPATTER) is able to accurately parse long sentences from a broad domain such as the WSJ. The probability model underlying the parser is a conditional, history-based model (Black, Jelinek, Lafferty, Magerman, Mercer, and Roukos 1993), where a parse tree is represented as a sequence of decisions that have been used to build the tree bottom-up. The probability of a decision is based on certain elements of the structure built up to that point. There is no hand-coded grammar underlying the parser, and the allowed moves of the parser (with their corresponding probabilities) are learned automatically from a treebank using decision trees.

An important feature of the model is that it is lexicalised, in the sense that the non-terminal nodes in a tree are labelled with the head word of the corresponding constituent, as well as the non-terminal label. Thus many of the moves made by the parser directly reflect the lexical dependencies present in the resulting tree. This is in contrast to many of the parsers built before SPATTER, which simply used part of speech tags as input. Collins reports that SPATTER performs at around 84% precision and recall on Section 23 of the Penn Treebank, compared with around 74% for a non-lexicalised PCFG (Charniak 1997).

[11] Some approaches we do not consider are the following: (Brill 1993; Eisner 1996a, 1996b; Ratnaparkhi 1997, 1999; Goodman 1997; Bod 1998).

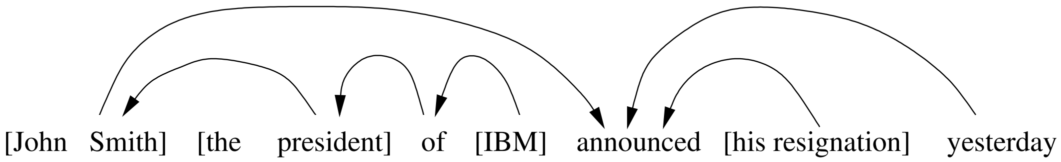
[John Smith] [the president] of [IBM] announced [his resignation] yesterday

Figure 2.3: Example dependencies and base NPs

Collins 1996, 1997

Collins (1996) was motivated by the work of Magerman, but introduced a model that is simpler and easier to train than the model underlying SPATTER. Collins' model is dependency based, and treats a parse as a set of baseNPs and a set of dependencies, as shown in Figure 2.3 (using an example from Collins 1996). The base NPs are shown as the bracketed constituents, where a base NP is an NP for which none of its child constituents are NPs; the dependencies are shown as directed links. The links are labelled in Collins' model, and correspond roughly to relations such as subject and object. The labels are derived automatically from the non-terminals in the parse tree.

Given a parse, T, represented as a set of baseNPs, B, and a set of dependencies, D, a conditional probability for T can be defined as follows:

$$p(T|S) = p(B,D|S) = p(B|S) \times p(D|S,B) \tag{2.27}$$

The method Collins uses for estimating $p(B|S)$ is based on ideas in Church 1988, where the detection of baseNPs is essentially seen as a tagging problem. The elements that need tagging are the gaps between words, which are classified as either at the start or end of a baseNP, between two adjacent baseNPs, or none of these. The probability of classifying the gap in a particular way is conditioned on the words and part of speech tags either side of the gap. A backing off approach is used to estimate the probabilities, where the backing off is from words to tags.

The probability model assumes that the dependencies are independent, and, since each word modifies exactly one other word (apart from the head of the sentence which modifies nothing), the probabilities of the dependencies can be multiplied together to give a probability for D. Collins' estimate for the probability that two word tag pairs $\langle w_i, t_i \rangle$ and $\langle w_j, t_j \rangle$ appear in relation R is based on the following relative frequency:

$$\hat{F}(R|\langle w_i, t_i \rangle, \langle w_j, t_j \rangle) = \frac{C(R, \langle w_i, t_i \rangle, \langle w_j, t_j \rangle)}{C(\langle w_i, t_i \rangle, \langle w_j, t_j \rangle)} \tag{2.28}$$

That is, the number of times that the pairs appear together in relation R is divided by the number of times that the pairs appear together in some sentence. These relative frequencies are smoothed, by backing off to counts based on tags. They are also normalised so that, for each word in the sentence, the probability of that word modifying one other word in the sentence, by some relation, is one. A further complication is that the estimates also incorporate a notion of the distance between the two dependents, which is found to greatly improve the performance of the model.

The parser itself is a simple bottom up chart parser, together with a beam search to increase the speed of the parser. The final results were at least as good as those for SPATTER, giving labelled precision and recall figures of around 85% using the standard training and test sets from the Penn Treebank.

Collins 1997 builds on Collins 1996, moving from a conditional model, $p(T|S)$, to a joint model, $p(T,S)$. The modelling is history-based, using a sequence of decisions to generate a parse tree in a top-down fashion. A series of models is presented, each new model increasing in complexity, and providing improved performance at each step. The models are lexicalised, in the sense that each node in a parse tree is labelled with the head word of the corresponding constituent, and this leads to a head-centred derivation of the tree. In the first model, the modifiers of a head (complements and adjuncts) are generated using two Markov processes, one for the modifiers to the left of the head, and one for the modifiers to the right. The second model introduces subcategorisation frames, so that the left and right halves of a frame are generated before generating the adjuncts. And finally, a third model attempts to incorporate Wh-movement, using the traces in the Penn Treebank for estimating the relevant probabilities.

These models improved on Collins' conditional model in a number of ways. First, the joint models generate only legal parses, whereas the conditional model is deficient, losing some probability mass to illegal dependency structures (such as structures with crossing dependency links). Also, the conditional model has no treatment of subcategorisation or Wh-movement (although Collins argues that the distance measure approximates subcategorisation to some extent). These improvements in the joint model led to a significant improvement in performance, the first model giving around 88% labelled precision and recall on the Penn Treebank test set (for sentences with 40 words or less), increasing to nearly 89% for model 3.

The parse selection system presented in Chapter 5 is in some ways motivated by the work of Collins. The system is based on lexical sense preferences, which are very similar to the lexical dependencies in Collins' models (as was argued in Chapter 1). In addition, structures corresponding to each possible parse are generated using a top down, head-centred stochastic process. However, the structures that are generated are dependency structures, rather than phrase structure trees. Dependency structures are used because we wanted to define the probability model in terms of parameters which, as far as possible, could be estimated using the method developed in Chapter 3. A possible advantage of our approach is that it can potentially be used with any parser that is able to produce output in the form of head dependencies. In contrast, the models in Collins 1997 (and many of the other current statistical parsing models) have been designed for parsers that produce trees in the style of the Penn Treebank.

Other generative parsing models

Charniak (2000), building on Charniak 1997, also describes a generative parsing model, in the style of Collins 1997. A difference is that Charniak uses maximum entropy modelling to estimate the parameters. The results reported by Charniak were 90.1% precision and recall (for sentences up to 40 words in length), which represented a significant increase over previous results, and at the time of writing are the best published results for this task. Collins (1999) comments that the performance of his models is likely to be improved by making use of estimation techniques from machine learning, such as maximum entropy (Ratnaparkhi 1999) or decision trees (Magerman 1995).

Bikel (2000) has recently defined a parsing model that also performs word sense disambiguation, using senses from WordNet. The model is a generative model, again in the style of Collins 1997. Word senses are integrated into the model by assuming that they are simply something else to be generated, along with the lexical items. A sense is generated first, and then a word to denote that sense, which means that probabilities of words conditioned on senses need to be estimated. Classes from WordNet are used to aid the estimation; however, the estimation method is different from that described in Chapter 3, in that a single class is not determined for each sense, but rather all the classes dominated by a hypernym of the sense are considered. Probabilities of the word conditioned on each class are then combined using linear interpolation. The training data were taken from SemCor (Miller, Leacock, Tengi, and Bunker 1993), which is a subset of the Penn Treebank that has been annotated with WordNet senses. The performance of the model can be measured without the WordNet senses, and it was found that adding the senses had no impact on the performance, although the work is still at an exploratory stage.

Other approaches to statistical parsing

Briscoe and Carroll (1993) define a probability model based on the moves of an LR parser (see also Briscoe and Carroll 1995, Carroll and Briscoe 1996, Carroll, Minnen, and Briscoe 1998). The grammar underlying the parser is a hand-written phrase structure grammar. The probability model is structural, and does not account for the probabilities of lexical dependencies. However, more context is taken into account than a PCFG, since the history that is considered at each parsing decision is conditional on the LR state, which can encode information in addition to the non-terminal being expanded. A dependency-based evaluation in Carroll, Minnen, and Briscoe 1999 shows that the latest version of the parsing system can identify some grammatical relations (such as subject and direct object) with high accuracy, but is less successful with other relations (such as the second object in a ditransitive construction and indirect object). The accurate identification of some relations, such as those corresponding to PP-attachment, is likely to require a more lexicalised probability model.

A current version of the Briscoe and Carroll parser is used throughout this thesis. The parser is highly robust, and has been used to provide large amounts of training data for the experiments reported in Chapters 5 and 6. It was also used for the parse selection experiments in Chapter 5, in order to provide the possible parses for a set of test sentences. A feature of the latest version is that the output is in the form of head dependency relations, which were used to create a dependency structure for each possible parse. In addition, the performance of the parser provided a useful benchmark against which to measure the performance of the dependency model.

Hektoen (1997) defines a probability model over logical forms, rather than syntactic structures, arguing that semantic relations are the key to accurate parse selection. A hand-written grammar was developed especially for this work, so that the requisite logical forms could be derived. A further novel aspect of the approach is that Bayesian estimation is used to estimate the parameters. Hektoen did attempt a direct comparison with SPATTER and Collins' conditional model, although the use of a hand-written grammar meant that only a subset of sentences from the Penn Treebank could be parsed. Also, Hektoen argues that the Parseval measures are not very suitable for his system, since they measure the ability of the system to reproduce the bracketing in the Treebank, which his system was not designed to do. The overall results were promising if not conclusive.

The parse selection system presented in Chapter 5 shares some similarities with Hektoen's. Hektoen's model is based on lexical cooccurrences in logical forms, which are similar to the dependencies we consider. Also, the dependency structures that are described in Chapter 5 exist somewhere between syntax and semantics, involving some 'deep structure' relations, such as passive subject. However, our dependency structures do not contain as much semantic information as Hektoen's logical forms (the logical forms contain quantifiers, for example). Finally, Hektoen's system could in principle be applied to any parser that can produce semantic forms, and is not restricted to parsers that have been designed to produce Penn Treebank style trees. The system in Chapter 5 shares a similar property.

Future directions for statistical parsing

Current work on statistical parsing is moving towards integrating statistical models with more linguistically motivated grammars, including unification-based grammars such as LFG and HSPG (Abney 1997; Johnson, Geman, Canon, Chi, and Riezler 1999; Riezler et al. 2000), TAG (Chiang 2000), and CCG (Hockenmaier, Bierner, and Baldridge 2000). It is hoped that the use of more sophisticated linguistic analyses will lead to probability models that are better able to discriminate between good and bad parses. A further advantage is that linguistically motivated grammars are able to produce logical forms, which can be a more useful output than phrase structure trees.

Chapter 3

Class-based Probability Estimation: how to select a suitable class

3.1 Problem specification

The problem addressed in this chapter is how to estimate $p(c|v,r)$, where c is a sense in a semantic hierarchy, v is a predicate and r is an argument position. The term 'predicate' is used loosely here, in that the predicate does not have to be a semantic object, but can simply be a word form. The kinds of 'predicates' considered in the later implementation chapters are verbs and adjectives, and no distinction is made between the different senses of a particular verb or adjective. The kinds of argument positions considered are the usual syntactic relations, such as subject, direct object, indirect object and so on.

The reason for not being too specific about the interpretation of 'predicate' and 'argument position' is that the estimation problem simply assumes we have a multi-set of nouns (or noun senses): $\{n_1, n_2, \ldots, n_k\}$. But note that the estimation technique is unlikely to work for any distribution over noun senses. The assumption underlying the technique is that the probability of a sense can be approximated by a probability based on a suitably chosen class. Thus, in practice, the technique is applied to distributions that are likely to satisfy that assumption, such as the argument slots of verbs. For example, it seems reasonable to suppose that the probability of $\langle\texttt{beefburger}\rangle$ appearing as an object of *eat* can be approximated, in some way, using a class such as FOOD.

Before describing the estimation technique, a precise description of the hierarchy is given. A brief description of the hierarchy has already been given in Chapter 2, and some of that description is repeated below.

3.1.1 The semantic hierarchy

The semantic hierarchy is the noun hypernym hierarchy of WordNet (Fellbaum 1998b), version 1.6. The hierarchy consists of senses, or what Miller (1998) calls *lexicalised concepts*, organised by the 'is-a-kind-of' relation. We follow Miller in using *concept* as short for *lexicalised concept*, and use the terms *concept* and *sense* interchangeably. It is important to realise that *concept* is being used in this way.

There are other taxonomies in WordNet, such as a verb taxonomy and adjective taxonomy, but only the noun taxonomy is used here. Hence, from now on, any reference to concepts in WordNet will mean concepts in the noun taxonomy only. Let $\mathcal{C} = \{c_1, \ldots, c_k\}$ be the set of concepts in WordNet. There are around $66,000$ different concepts in version 1.6. Each concept is represented by a *synonym set* (or *synset*), which is the set of synonymous words that can be used to denote that concept. For example, the synset for the concept $\langle\texttt{taxi}\rangle$ is $\{\mathit{cab}, \mathit{hack}, \mathit{taxi}, \mathit{taxicab}\}$. Let $\mathcal{N}$ be the set of nouns appearing in synsets in WordNet; we use $\mathsf{syn}(c) \subseteq \mathcal{N}$ to denote the synset for

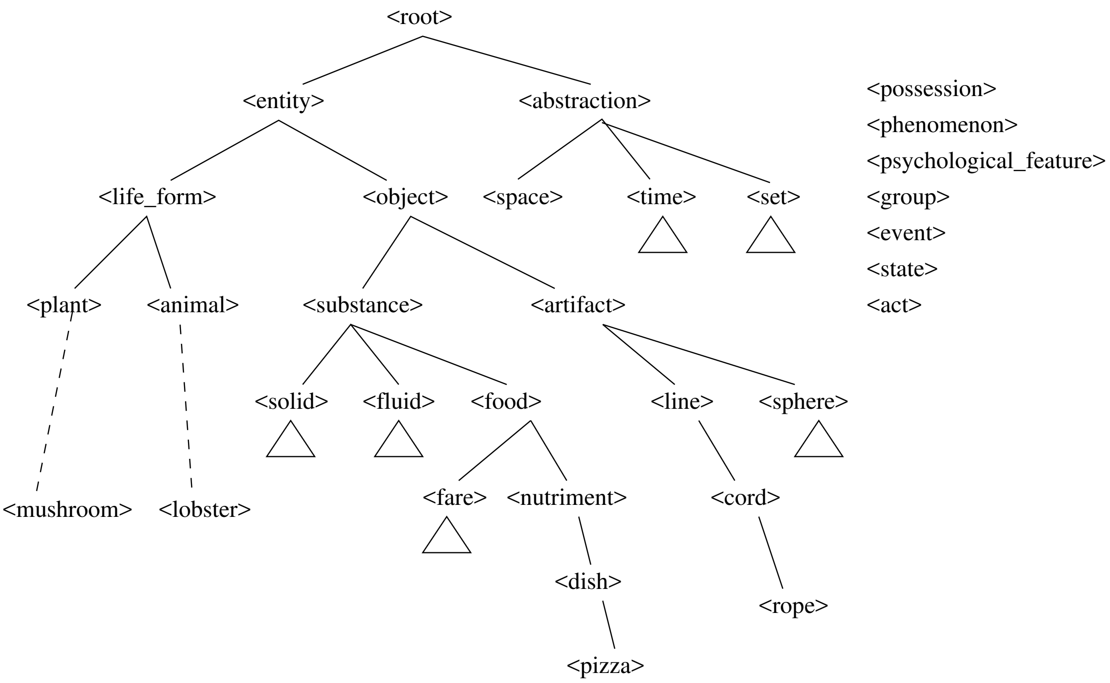

Figure 3.1: Part of the WordNet hierarchy

concept c, and $\mathsf{cn}(n) = \{\, c \,|\, n \in \mathsf{syn}(c) \,\}$ to denote the set of concepts that can be denoted by the noun n.

The hierarchy has the structure of a directed acyclic graph (although only around one percent of the nodes in the graph have more than one parent), where the edges of the graph constitute what we call the '$\mathsf{direct-isa}$' relation. Let $\mathsf{isa} \subseteq \mathcal{C} \times \mathcal{C}$ be the transitive, reflexive closure of $\mathsf{direct-isa}$, then c' isa c implies c' is a kind of c. If c' isa c, then c is a *hypernym* of c' and c' is a *hyponym* of c. In fact, the hierarchy is not a single hierarchy, but consists of nine separate sub-hierarchies. The sub-hierarchies are headed by the most general kind of concept, and the roots of the sub-hierarchies are shown in Figure 3.1, which shows part of the WordNet hierarchy. (The seven roots in addition to $\langle\texttt{entity}\rangle$ and $\langle\texttt{abstraction}\rangle$ are shown as a list.) Only a small selection of children for each node are shown, and dashed lines and triangles indicate that part of the hierarchy is missing from the figure. For the purposes of this work we add a common root dominating the nine sub-hierarchies, which we denote $\langle\texttt{root}\rangle$. The concept $\langle\texttt{root}\rangle$ can be thought of as having the empty set as a synset.

There are some important points of clarification regarding the hierarchy. First, every concept has a non-empty synset (except the notional concept $\langle\texttt{root}\rangle$). Even the most general concepts, such as $\langle\texttt{entity}\rangle$, can be denoted by some noun; the synset for $\langle\texttt{entity}\rangle$ is $\{\, \textit{entity}, \textit{something} \,\}$. Second, there is an important distinction between an individual concept and a set of concepts. For example, the concept $\langle\texttt{entity}\rangle$ should not be confused with the set or class consisting of kinds of entities. To make this distinction clear, we use $\overline{c} = \{\, c' \,\underline{|\, c' \ \mathsf{isa\ c} \,\}}$ to denote the set of concepts dominated by concept c, including c itself. For example, $\overline{\langle\texttt{animal}\rangle}$ is the set consisting of those concepts corresponding to kinds of animals (including $\langle\texttt{animal}\rangle$ itself).

We can now be more precise about the probability $p(c|v,r)$, which is to be interpreted as follows: this is the conditional probability that some noun n in $\mathsf{syn}(c)$, when denoting concept c, appears in position r of predicate v (given r and v). In order to simplify the discussion in the rest of the chapter, it is assumed that v is a verb and that r ranges over the verbal 'slots' subject, direct object and so on; however, it should be remembered that the estimation procedure can be applied to any predicate that takes nominal arguments. In Chapter 5, the procedure is applied to non-verbal predicates such as adjectives as well as verbs.

3.2 Class-based probability estimation

This section explains how a set of concepts, or class, from WordNet can be used to estimate the probability of an individual concept. More specifically, we explain how a set of concepts $\overline{c'}$, where c' is some hypernym of concept c, can be used to estimate $p(c|v,r)$. (Recall that $\overline{c'}$ denotes the set of concepts dominated by c', including c' itself.) The example used throughout this section is $p(\langle\texttt{dog}\rangle|\mathit{run},\mathrm{subj})$. One possible approach would be to simply substitute $\overline{c'}$ for the individual concept c, so the class $\overline{\langle\texttt{animal}\rangle}$ might be substituted for the concept $\langle\texttt{dog}\rangle$, for example. However, this is a poor solution, since $p(\overline{c'}|v,r)$ is the conditional probability that some noun denoting a concept in $\overline{c'}$ appears in position r of verb v. So $p(\overline{\langle\texttt{animal}\rangle}|\mathit{run},\mathrm{subj})$ is the probability that some noun denoting a kind of animal appears in the subject position of the verb *run*. Probabilities of sets of concepts are obtained by summing over the concepts in the set:

$$p(\overline{c'}|v,r) = \sum_{c''\in\overline{c'}} p(c''|v,r) \tag{3.1}$$

This means that $p(\overline{\langle\texttt{animal}\rangle}|\mathit{run},\mathrm{subj})$ is likely to be much greater than $p(\langle\texttt{dog}\rangle|\mathit{run},\mathrm{subj})$, and not a good approximation of $p(\langle\texttt{dog}\rangle|\mathit{run},\mathrm{subj})$.

The proposal in response to this is to invert the relevant probability using Bayes theorem and *condition* on sets of concepts. If it can be shown that $p(v|\overline{c'},r)$, for some hypernym c' of c, is a reasonable approximation of $p(v|c,r)$, then we have a way of estimating $p(c|v,r)$. The probability $p(v|c,r)$ is obtained from $p(c|v,r)$ as follows:

$$p(c|v,r) = p(v|c,r)\frac{p(c|r)}{p(v|r)} \tag{3.2}$$

Since the probabilities $p(c|r)$ and $p(v|r)$ are conditioned on the argument slot only, it is more likely that these can be estimated satisfactorily using relative frequency estimates. Alternatively, a standard smoothing technique such as Good-Turing could be used.[1]

This only leaves $p(v|c,r)$. Continuing with the *dog* example, the proposal is to estimate $p(\mathit{run}|\langle\texttt{dog}\rangle,\mathrm{subj})$ using a relative frequency estimate of $p(\mathit{run}|\overline{\langle\texttt{animal}\rangle},\mathrm{subj})$, or an estimate based on a similar, suitably chosen class. In Figure 3.2, it is shown that if $p(v|c'',r)$ is the same for each c'' in $\overline{c'}$, where c' is some hypernym of c, then $p(v|\overline{c'},r)$ will be equal to $p(v|c,r)$:

$$p(v|c'',r) = k \text{ for all } c'' \in \overline{c'} \;\Rightarrow\; p(v|\overline{c'},r) = k \tag{3.3}$$

Proposition 3.3 suggests a way of deciding if $p(v|\overline{c'},r)$ is likely to be a useful approximation of $p(v|c,r)$: compare estimates of the probabilities $p(v|c'',r)$. If the estimates are very different, then it is unlikely that $p(v|\overline{c'},r)$ will be a good approximation of $p(v|c,r)$. However, there is a problem with this suggestion: sparse data problems mean that relative frequency estimates of the probabilities $p(v|c'',r)$ are likely to be unreliable. A more promising approach would be to compare probabilities conditioned on *sets* of concepts; that way, the estimates of the probabilities being compared would be more reliable.

We are able to derive such an approach by assuming that the hierarchy is a tree. In the tree case, if $p(v|\overline{c'_i},r) = k$, for each child c'_i of c', and $p(v|c',r) = k$, then it can be shown that $p(v|\overline{c'},r)$ will also be equal to k:

$$p(v|\overline{c'_i},r) = k \text{ for all children } c'_i \text{ of } c', \text{ and } p(v|c',r) = k \;\Rightarrow\; p(v|\overline{c'},r) = k \tag{3.10}$$

Note that now we are dealing with probabilities conditioned on *sets* of concepts: $p(v|\overline{c'_i},r)$, and so sparse data will be less of a problem. (In practice, we ignore the probability $p(v|c',r)$, and

[1] Unsmoothed estimates were used in the work described in this thesis.

$$\begin{align}
p(v|\overline{c'},r) &= p(\overline{c'}|v,r)\frac{p(v|r)}{p(\overline{c'}|r)} \tag{3.4}\\
&= \frac{p(v|r)}{p(\overline{c'}|r)}\sum_{c''\in\overline{c'}} p(c''|v,r) \tag{3.5}\\
&= \frac{p(v|r)}{p(\overline{c'}|r)}\sum_{c''\in\overline{c'}} p(v|c'',r)\frac{p(c''|r)}{p(v|r)} \tag{3.6}\\
&= \frac{1}{p(\overline{c'}|r)}\sum_{c''\in\overline{c'}} k\, p(c''|r) \tag{3.7}\\
&= \frac{k}{p(\overline{c'}|r)}\sum_{c''\in\overline{c'}} p(c''|r) \tag{3.8}\\
&= k \tag{3.9}
\end{align}$$

Figure 3.2: Proof of proposition 3.3

compare the probabilities $p(v|\overline{c'_i},r)$ only.) The proof of proposition 3.10 is given in Figure 3.3, and is explained in detail below.

The first line (3.12) applies Bayes theorem to the probability $p(v|\overline{c'},r)$. Line 3.13 rewrites the probability $p(\overline{c'}|v,r)$ as the sum of the probabilities of the sets dominated by the daughters of c', $\sum_i p(\overline{c'_i}|v,r)$, plus the probability of c' itself, $p(c'|v,r)$. This equality holds because the probability of a set of concepts, $p(\overline{c'}|v,r)$, has been defined in 3.1 as the sum of the probabilities of the concepts in the set. However, note that the equality only holds in the tree case, and this is where the proofs in Figures 3.2 and 3.3 differ. For a DAG, the probability of a set of concepts dominated by c' cannot be obtained by summing the probabilities of the sets dominated by the daughters of c' (plus the probability of c' itself). The reason is that, in the sum $\sum_i p(\overline{c'_i}|v,r)$, the probabilities of some individual concepts in a DAG can be counted more than once. This occurs when two of the children of c' share a common child, in which case the following could hold:

$$\sum_i p(\overline{c'_i}|v,r) + p(c'|v,r) > p(\overline{c'}|v,r) \tag{3.11}$$

This cannot occur in the tree case, since different nodes cannot share children.

Returning to the proof, line 3.14 applies Bayes theorem once more to the probabilities appearing in the brackets in 3.13. Line 3.15 follows from 3.14 because the terms $p(v|r)$ cancel, and, by hypothesis, $p(v|\overline{c'_i},r) = k$ for each daughter c'_i, and $p(v|c',r) = k$. Finally, the bracketed term in 3.16 is equal to $p(\overline{c'}|r)$, by 3.1 and the assumption of a tree.

Proposition 3.10 is useful because it shows how probabilities conditioned on sets of concepts can remain constant when moving up the hierarchy. This suggests a way of finding a suitable set, $\overline{c'}$, for concept c: initially set c' equal to c and move up the hierarchy until there is a significant change in $p(v|\overline{c'},r)$. Estimates of $p(v|\overline{c'_i},r)$, for each child c'_i of c', can be compared to see if $p(v|\overline{c'},r)$ has significantly changed. We cannot expect the $p(v|\overline{c'_i},r)$ to be exactly the same, which proposition 3.10 strictly requires, but if the estimates indicate that the $p(v|\overline{c'_i},r)$ are similar, then we can expect that $p(v|\overline{c'},r)$ has not changed significantly from the previous node. We also require the initial assumption that $p(v|\overline{c},r)$ is close to $p(v|c,r)$. (In fact, $p(v|\overline{c},r)$ is equal to $p(v|c,r)$ for the case when c is a leaf node.) This assumption is needed because the procedure checks that a probability conditioned on a *set* of concepts remains unchanged, whereas the aim is to estimate a probability conditioned on a single concept: $p(v|c,r)$.

This procedure does have the disadvantage that it applies to a tree, whereas WordNet is a DAG; however, since WordNet is a close approximation to a tree, in that only around 1% of the nodes have more than one parent, we do not expect this to be a problem. The procedure for finding a

$$
\begin{aligned}
p(v|\overline{c'},r) &= p(\overline{c'}|v,r)\frac{p(v|r)}{p(\overline{c'}|r)} && (3.12)\\
&= \frac{p(v|r)}{p(\overline{c'}|r)}\left(\sum_i p(\overline{c'_i}|v,r) + p(c'|v,r)\right) && (3.13)\\
&= \frac{p(v|r)}{p(\overline{c'}|r)}\left(\sum_i p(v|\overline{c'_i},r)\frac{p(\overline{c'_i}|r)}{p(v|r)} + p(v|c',r)\frac{p(c'|r)}{p(v|r)}\right) && (3.14)\\
&= \frac{1}{p(\overline{c'}|r)}\left(\sum_i k\,p(\overline{c'_i}|r) + k\,p(c'|r)\right) && (3.15)\\
&= \frac{k}{p(\overline{c'}|r)}\left(\sum_i p(\overline{c'_i}|r) + p(c'|r)\right) && (3.16)\\
&= k && (3.17)
\end{aligned}
$$

Figure 3.3: Proof of proposition 3.10

suitable set is described in more detail in Section 3.4, after the test for comparing the probabilities has been described in Section 3.3.

Finally, we note that the proposed estimation method does not guarantee that the estimates form a probability distribution over the concepts in the hierarchy, and so a normalisation factor is required:

$$
p_{sc}(c|v,r) = \frac{\hat{p}(v|[c,v,r],r)\frac{\hat{p}(c|r)}{\hat{p}(v|r)}}{\sum_{c'\in\mathcal{C}}\hat{p}(v|[c',v,r],r)\frac{\hat{p}(c'|r)}{\hat{p}(v|r)}} \tag{3.18}
$$

We use p_{sc} to denote an estimate obtained using our method (since the technique finds sets of semantically similar senses, or '**S**imilarity **C**lasses'), and $[c,v,r]$ to denote the class chosen for concept c in position r of verb v; $\hat{p}$ denotes a relative frequency estimate, and $\mathcal{C}$ denotes the set of concepts in the hierarchy.

Before describing the generalisation procedure in more detail, the next section considers the problem of ambiguous data.

3.2.1 Estimating the relevant probabilities

The data used to estimate the relevant probabilities are assumed to be (n,v,r) triples: a noun, verb and argument position. Such data can be obtained from a treebank or from a shallow parser. It is assumed that each use of a noun in the data corresponds to exactly one concept. The relative frequency estimates for the probabilities used to estimate $p(c|v,r)$ are as follows:

$$
\hat{p}(c|r) = \tfrac{f(c,r)}{f(r)} = \frac{\sum_{v'\in\mathcal{V}} f(c,v',r)}{\sum_{v'\in\mathcal{V}}\sum_{c'\in\mathcal{C}} f(c',v',r)} \tag{3.19}
$$

$$
\hat{p}(v|r) = \tfrac{f(v,r)}{f(r)} = \frac{\sum_{c'\in\mathcal{C}} f(c',v,r)}{\sum_{v'\in\mathcal{V}}\sum_{c'\in\mathcal{C}} f(c',v',r)} \tag{3.20}
$$

$$
\hat{p}(v|\overline{c'},r) = \tfrac{f(\overline{c'},v,r)}{f(\overline{c'},r)} = \frac{\sum_{c''\in\overline{c'}} f(c'',v,r)}{\sum_{v'\in\mathcal{V}}\sum_{c''\in\overline{c'}} f(c'',v',r)} \tag{3.21}
$$

where $f(c,v,r)$ is the number of (n,v,r) triples in the data in which n is being used to denote c, and $\mathcal{V}$ is the set of verbs in the data. A problem arises because $f(c,v,r)$ is defined in terms of noun

$\overline{c_i}$	$\hat{f}(\overline{c_i}, run, \mathsf{subj})$	$\hat{f}(\overline{c_i}, \mathsf{subj})$ $-\hat{f}(\overline{c_i}, run, \mathsf{subj})$	$\hat{f}(\overline{c_i}, \mathsf{subj}) =$ $\sum_{v \in \mathcal{V}} \hat{f}(\overline{c_i}, v, \mathsf{subj})$
$\overline{\langle \texttt{bitch} \rangle}$	0.3 (0.5)	26.7 (26.6)	27.0
$\overline{\langle \texttt{dog} \rangle}$	12.8 (10.5)	620.4 (622.7)	633.2
$\overline{\langle \texttt{wolf} \rangle}$	0.3 (0.6)	38.7 (38.4)	39.0
$\overline{\langle \texttt{jackal} \rangle}$	0.0 (0.3)	20.0 (19.7)	20.0
$\overline{\langle \texttt{wild\_dog} \rangle}$	0.0 (0.0)	3.0 (3.0)	3.0
$\overline{\langle \texttt{hyena} \rangle}$	0.0 (0.2)	10.0 (9.8)	10.0
$\overline{\langle \texttt{fox} \rangle}$	0.0 (1.2)	72.3 (71.1)	72.3
	13.4	791.1	804.5

Table 3.1: Contingency table for the children of $\langle \texttt{canine} \rangle$ in the subject position of *run*

senses, but the data consist of nouns. For now, a simple approach is taken which is to estimate $f(c, v, r)$ by distributing the count for each noun n in $\mathsf{syn}(c)$ evenly among all senses of the noun:

$$\hat{f}(c, v, r) = \sum_{n \in \mathsf{syn}(c)} \frac{f(n, v, r)}{|\mathsf{cn}(n)|} \tag{3.22}$$

where $|\mathsf{cn}(n)|$ is the cardinality of $\mathsf{cn}(n)$. This approach is taken by Resnik (1998), Li and Abe (1998), Ribas (1995b) and McCarthy (1997). Resnik explains how this apparently crude technique works surprisingly well. Resnik's explanation is discussed in Chapter 4, where a novel alternative is described; for the purposes of this chapter, it is assumed that splitting the count equally is an adequate solution.

3.3 Using a chi-squared test to compare probabilities

The test used to compare the $p(v|\overline{c_i'}, r)$ is a chi-squared test. Continuing with the example of $\langle \texttt{dog} \rangle$ in the subject position of *run*, consider the problem of deciding if $p(run|\overline{\langle \texttt{canine} \rangle}, \mathsf{subj})$ is a good approximation of $p(run|\overline{\langle \texttt{dog} \rangle}, \mathsf{subj})$. (The concept $\langle \texttt{canine} \rangle$ is the parent of $\langle \texttt{dog} \rangle$ in WordNet.) To do this, we compare the probabilities $p(run|\overline{c_i'}, \mathsf{subj})$, where the c_i' are the children of $\langle \texttt{canine} \rangle$. First, a null hypothesis is formulated. In this case, the null hypothesis is that the probabilities $p(run|\overline{c_i'}, \mathsf{subj})$ are the same for each child c_i'. By judging the strength of the evidence against the null hypothesis, it can be determined how similar the true probabilities are likely to be.

The next stage, after formulating a null hypothesis, is to create a contingency table. The table contains a row for each child, c_i', and a number of columns. One column contains counts arising from concepts in $\overline{c_i'}$ appearing in the subject position of *run*: $\hat{f}(\overline{c_i'}, run, \mathsf{subj})$. A second column contains counts arising from concepts in $\overline{c_i'}$ appearing in the subject position of a verb other than *run*. The totals for each row and column also appear in the table; these are known as the *marginal totals*. An example contingency table, based on counts obtained from a subset of the BNC, is given in Table 3.1. (Recall that the frequencies are estimated by distributing the count for a noun equally among the noun's senses: this explains the fractional counts.) The data leading to the counts were extracted using the system of Briscoe and Carroll (1997). All of the data used in this chapter were obtained using that system.

The figures in brackets are the expected values, *given that the null hypothesis is true*. The expected values are obtained from the marginal totals. For example, the expected value 10.5 (in the $\langle \texttt{dog} \rangle$ row) is calculated from the marginal totals as follows: $10.5 = 633.2 \times 13.4/804.5$. Similarly, 26.6 (in the $\langle \texttt{bitch} \rangle$ row) $= 27.0 \times 791.1/804.5$, and 0.2 (in the $\langle \texttt{hyena} \rangle$ row) $= 10.0 \times 13.4/804.5$. So to obtain some indication of how likely the null hypothesis is to be false, we can compare the actual values with the expected values in the table. The actual and expected values are compared using a chi-squared statistic. The statistic that usually appears in text books is the Pearson chi-squared statistic, denoted X^2:

$$X^2 = \sum_{i,j} \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \tag{3.23}$$

where o_{ij} is the observed value for the cell in row i and column j, and e_{ij} is the corresponding expected value. An alternative statistic, but less well known, is the log-likelihood chi-squared statistic, denoted G^2:

$$G^2 = 2\sum_{i,j} o_{ij} \ln \frac{o_{ij}}{e_{ij}} \tag{3.24}$$

where ln is log to base e.

A reference to a chi-squared test will often mean a test employing X^2, and the 'Pearson' is left implicit. Regarding the question of which statistic to use, X^2 and G^2 have similar values when the counts in the table are large (Agresti 1996). However, the statistics behave differently when the contingency table contains low counts, and, since corpus data is likely to lead to some low counts, the question is an important one. Dunning (1993)[2] argues that G^2 is more suitable for corpus-based NLP than X^2. However, in Section 3.6 we dispute Dunning's claim, and the question of whether to use G^2 or X^2 will be discussed further there. For now, we continue with the discussion of how the value of either statistic can be used to give an indication of the strength of evidence against the null hypothesis.

The reason these statistics are useful is that, if the null hypothesis is true, then the sampling distributions of X^2 and G^2 are approximately chi-squared distributions; that is, if the chi-squared test were repeated many times, the different values of either statistic would be approximately distributed according to a chi-squared distribution. In fact, there are a number of chi-squared distributions, and any particular distribution is specified by the *degrees of freedom.* This is calculated from the contingency table as $(r-1)(c-1)$, where r is the number of rows and c is the number of columns. For Table 3.1, the relevant chi-squared distribution has $(7-1)(2-1) = 6$ degrees of freedom. So to decide on the strength of evidence against the null hypothesis, we can see where the value for the statistic lies in the chi-squared distribution. If it lies in an 'unlikely' region, this is taken as evidence that the null hypothesis is false, and leads to the null hypothesis being rejected. A rejection of the kind of null hypothesis being considered here is taken to imply the following: that the probabilities being compared are not similar enough for the relevant class-based estimate to be a useful estimate. If the null hypothesis corresponding to Table 3.1 were rejected, the conclusion would be that $p(run|\overline{\langle\texttt{canine}\rangle},\mathrm{subj})$ is not a good approximation of $p(run|\overline{\langle\texttt{dog}\rangle},\mathrm{subj})$.

The question then becomes how unlikely the 'unlikely region' should be. Traditionally, scientists using the chi-squared test have chosen the tail of the distribution corresponding to a probability of 0.05 (or some similar value). This probability is known as the *significance level* of the test and is usually denoted α. So, if the null hypothesis is true, the probability of obtaining a G^2 or X^2 value in this region is approximately 0.05. The lower bound on the unlikely region is known as the *critical value.* If the chi-squared statistic exceeds the critical value, then this leads to a rejection of the null hypothesis.

For Table 3.1, the value of G^2 is 3.8, and the value of X^2 is 2.5. The critical value corresponding to an α value of 0.05 and 6 degrees of freedom is 12.6. Thus, for an α value of 0.05, the null hypothesis would not be rejected for either statistic, and the conclusion would be that the probabilities are similar enough for $p(run|\overline{\langle\texttt{canine}\rangle},\mathrm{subj})$ to be a reasonable approximation of $p(run|\overline{\langle\texttt{dog}\rangle},\mathrm{subj})$. As a further example, Table 3.2 gives counts for the daughters of $\langle\texttt{liquid}\rangle$ in

[2] The formula for G^2 given in Dunning 1993 is a more complex version of the one given here, but the two are equivalent.

$\overline{c_i}$	$\hat{f}(\overline{c_i}, drink, \text{obj})$		$\hat{f}(\overline{c_i}, \text{obj})$ $-\hat{f}(\overline{c_i}, drink, \text{obj})$		$\hat{f}(\overline{c_i}, \text{obj}) =$ $\sum_{v \in \mathcal{V}} \hat{f}(\overline{c_i}, v, \text{obj})$
$\overline{\langle\texttt{beverage}\rangle}$	261.0	(238.7)	2,367.7	(2,390.0)	2,628.7
$\overline{\langle\texttt{supernatant}\rangle}$	0.0	(0.1)	1.0	(0.9)	1.0
$\overline{\langle\texttt{alcohol}\rangle}$	11.5	(9.4)	92.0	(94.1)	103.5
$\overline{\langle\texttt{ammonia}\rangle}$	0.0	(0.8)	8.5	(7.7)	8.5
$\overline{\langle\texttt{antifreeze}\rangle}$	0.0	(0.1)	1.0	(0.9)	1.0
$\overline{\langle\texttt{distillate}\rangle}$	0.0	(0.5)	6.0	(5.5)	6.0
$\overline{\langle\texttt{water}\rangle}$	12.0	(31.6)	335.7	(316.1)	347.7
$\overline{\langle\texttt{ink}\rangle}$	0.0	(2.9)	32.0	(29.1)	32.0
$\overline{\langle\texttt{liquor}\rangle}$	0.7	(1.1)	11.6	(11.2)	12.3
	285.2		2,855.5		3,140.7

Table 3.2: Contingency table for the children of ⟨`liquid`⟩ in the object position of *drink*

the object position of *drink*. Again, the counts have been obtained from a subset of the BNC. Not all the sets dominated by the daughters are shown, as some, such as $\overline{\langle\texttt{sheep\_dip}\rangle}$, never appear in the object position of a verb. This example is designed to show a case where the null hypothesis is rejected. The value of G^2 for this table is 29.0, and the value of X^2 is 21.2. So for G^2, even if an α value as low as 0.0005 were being used (for which the critical value is 27.9 for 8 degrees of freedom), the null hypothesis would still be rejected. For X^2, the null hypothesis is rejected for α values greater than 0.005. This seems reasonable, since probabilities associated with the daughters of ⟨`liquid`⟩ would be expected to be very different with regard to the object position of *drink*.

The relevant question at this point is how to decide on a value for α. One approach would be to just choose a value, such as 0.05. An alternative solution, which is adopted here, is to treat α as a parameter and set it empirically: take a held-out test set and choose the level of α that maximises the performance on the relevant task. In later chapters the estimation techniques are used to resolve PP–attachment ambiguities, using the now standard test and training set from Ratnaparkhi et al. 1994. There is also a development set that could be used to set α, by choosing the value that gives the best disambiguation performance on the development set. Note that this approach sets no constraints on the value of α: the value could be as high as 0.995 or as low as 0.0005, depending on the particular application. In Section 3.5 it is shown how the value of α affects the generalisation level, and, in later chapters, it is shown how the value affects performance on particular disambiguation tasks.

3.4 The procedure for determining an appropriate level of generalisation

In Section 3.2 a procedure was suggested for finding an appropriate class, $\overline{c'}$, to represent concept c in position r of verb v. We refer to $\overline{c'}$ as the 'similarity-class' of c with respect to v and r, and we refer to the hypernym c' as $\texttt{top}(c,v,r)$ (since the chosen hypernym sits at the 'top' of the similarity-class). The procedure works as follows. Initially, concept c is assigned to a variable `top`. Then, by working up the hierarchy, successive hypernyms of c are assigned to `top`, which continues until the probabilities associated with the sets of concepts dominated by `top` and the siblings of `top` are significantly different. Once a node is reached which results in a significant result for the chi-squared test, the procedure stops, and `top` is returned as $\texttt{top}(c,v,r)$. In cases where a concept has more than one parent, the parent is chosen which results in the lowest value of the chi-squared statistic, as this indicates the probabilities are more similar. The set $\overline{\texttt{top}(c,v,r)}$ is the similarity-class of c for verb v and position r. Figure 3.4 gives an algorithm for determining

```
Algorithm top(c, v, r):
top ← c
sig_result ← false
comment parent_min gives lowest $G^2$ value, $G^2_{min}$
while not sig_result & top ≠ ⟨root⟩ do
    $G^2_{min} \leftarrow \infty$
    for all parents of top do
        calculate $G^2$ for sets dominated by children of parent
        if $G^2 < G^2_{min}$
          then $G^2_{min} \leftarrow G^2$
               parent_min ← parent
    end
    if chi-squared test for parent_min is significant
      then sig_result ← true
    else move up to next node: top ← parent_min
end
return top
```

Figure 3.4: An algorithm for determining $\mathsf{top}(c,v,r)$.

$\mathsf{top}(c,v,r)$.

Figure 3.5 gives an example of the procedure at work. Here, $\mathsf{top}(\langle\texttt{soup}\rangle, \mathit{stir}, \mathrm{obj})$ is being determined. The example is based on data from a subset of the BNC, which had 303 cases of an argument in the object position of *stir*. The G^2 statistic is used, together with an α value of 0.05. Initially, top is set to $\langle\texttt{soup}\rangle$, and the probabilities corresponding to the children of $\langle\texttt{dish}\rangle$ are compared: $p(\mathit{stir}|\overline{\langle\texttt{soup}\rangle}, \mathrm{obj})$, $p(\mathit{stir}|\overline{\langle\texttt{lasagne}\rangle}, \mathrm{obj})$, $p(\mathit{stir}|\overline{\langle\texttt{haggis}\rangle}, \mathrm{obj})$ and so on for the rest of the children. The chi-squared test results in a G^2 value of 14.5, compared to a critical value of 55.8. Since G^2 is less than the critical value, the procedure moves up to the next node. This continues until a significant result is obtained, which first occurs at $\langle\texttt{substance}\rangle$ when comparing the children of $\langle\texttt{object}\rangle$. Thus $\langle\texttt{substance}\rangle$ is the chosen level of generalisation.

Before giving some example levels of generalisation, it is worth making some comparisons with the other WordNet approaches. First, note that we have not made a uniform distribution assumption, as Li and Abe do (equation 2.13). Furthermore, the problem described in Section 2.1.1, stemming from the fact that Li and Abe compare frequencies in order to generalise, does not arise. This problem is avoided because we compare probabilities conditioned on sets of concepts, rather than the frequencies of senses. And finally, the generalisation procedure is able to return a suitable class for arguments that are negatively associated with some predicate. (Section 2.1.1 explained how such arguments cause a problem for Resnik's approach.) To see why, consider applying the generalisation procedure to $\langle\texttt{location}\rangle$ in the object position of *eat*; the procedure is unlikely to get as high as $\langle\texttt{entity}\rangle$ (as we argued Resnik's approach is likely to do), since the probabilities corresponding to the daughters of $\langle\texttt{entity}\rangle$ are likely to be very different with respect to the object position of *eat*.

There is one disadvantage of our approach, at least compared with Li and Abe's use of MDL, which is that we are required to store the sense frequencies associated with every predicate and argument position. Li and Abe, in contrast, are only required to store the tree cut models associated with each predicate and argument position, which are likely to require much less space.

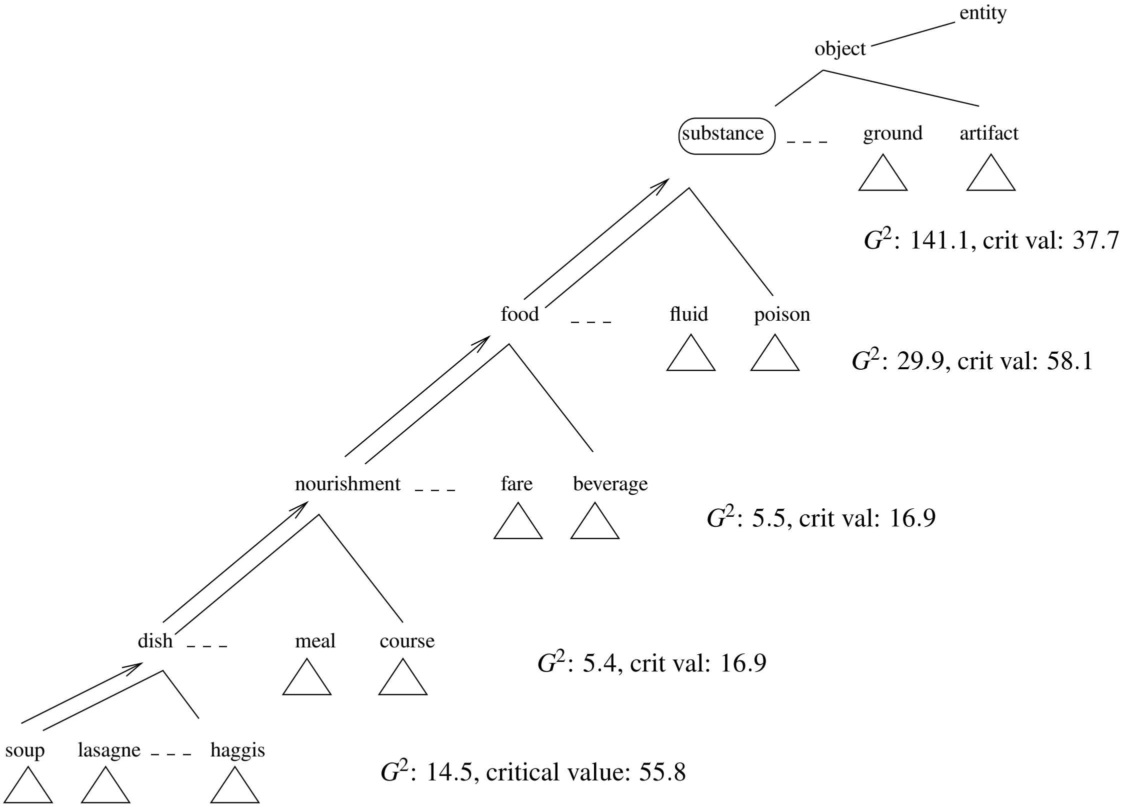
entity
object
substance
ground
artifact
G^2: 141.1, crit val: 37.7
food
fluid
poison
G^2: 29.9, crit val: 58.1
nourishment
fare
beverage
G^2: 5.5, crit val: 16.9
dish
meal
course
G^2: 5.4, crit val: 16.9
soup
lasagne
haggis
G^2: 14.5, critical value: 55.8

Figure 3.5: An example generalisation: determining $\mathtt{top}(\langle\mathtt{soup}\rangle, \mathit{stir}, \mathrm{obj})$

3.5 Example generalisation levels

In this section, we show how the level of generalisation varies with the value for α and how it varies with the size of the data set. A point of clarification is required before presenting the results. In other work on acquiring selectional preferences (Ribas 1995b; McCarthy 1997; Li and Abe 1998; Wagner 2000), the level of generalisation is determined for a small number of hand-picked verbs and the result compared with the researcher's intuition about the most appropriate level for representing a selectional preference. According to this approach, if $\langle\mathtt{sandwich}\rangle$ were chosen to represent $\langle\mathtt{beefburger}\rangle$ in the object position of *eat*, this might be considered an under-generalisation, since $\langle\mathtt{food}\rangle$ might be considered more appropriate. For this work we argue that such an evaluation is not appropriate and should be avoided (not least because it is subjective and can only be applied to a handful of cases). Since the purpose of this work is probability estimation, the most appropriate level is the one that leads to the most accurate estimate, and this may or may not agree with intuition. The purpose of this section is not to show that the acquired levels are 'correct,' but to show how the levels vary with α and sample size. Later chapters give objective task-based evaluations.

To show how the level of generalisation varies with changes in α, $\mathtt{top}(c, v, \mathrm{obj})$ was determined for a number of hand-picked (c, v, obj) triples over a range of values for α. The G^2 statistic was used in the chi-squared tests. The results are shown in Table 3.3. The triples were chosen to give a range of strongly and weakly selecting verbs and a range of verb frequencies. The number of times the verb occurred with some object is given in the table. The data were again extracted from a subset of the BNC using the system of Briscoe and Carroll (1997).

The results suggest that the generalisation level becomes more specific as α increases. This is to be expected, since, given a contingency table chosen at random, a higher value of α is more likely to lead to a significant result than a lower value of α. We also see that, for some cases, the value of α has little effect on the level. We would expect there to be less change in the level of

(c,v,r), $f(v,r)$	α	
(⟨coffee⟩, *drink*, obj)	0.0005	⟨coffee⟩⟨BEVERAGE⟩⟨food⟩...⟨object⟩⟨entity⟩
	0.05	⟨coffee⟩⟨BEVERAGE⟩⟨food⟩...⟨object⟩⟨entity⟩
$f(\textit{drink},\text{obj}) = 849$	0.5	⟨coffee⟩⟨BEVERAGE⟩⟨food⟩...⟨object⟩⟨entity⟩
	0.995	⟨coffee⟩⟨BEVERAGE⟩⟨food⟩...⟨object⟩⟨entity⟩
(⟨hotdog⟩, *eat*, obj)	0.0005	⟨hotdog⟩⟨sandwich⟩⟨snack_food⟩⟨DISH⟩...⟨food⟩...⟨entity⟩
	0.05	⟨hotdog⟩⟨sandwich⟩⟨snack_food⟩⟨DISH⟩...⟨food⟩...⟨entity⟩
$f(\textit{eat},\text{obj}) = 1,703$	0.5	⟨hotdog⟩⟨sandwich⟩⟨snack_food⟩⟨DISH⟩...⟨food⟩...⟨entity⟩
	0.995	⟨hotdog⟩⟨SANDWICH⟩⟨snack_food⟩⟨dish⟩...⟨food⟩...⟨entity⟩
(⟨Socrates⟩, *kiss*, obj)	0.0005	⟨Socrates⟩...⟨person⟩⟨life_form⟩⟨CAUSAL_AGENT⟩⟨entity⟩
	0.05	⟨Socrates⟩...⟨person⟩⟨life_form⟩⟨CAUSAL_AGENT⟩⟨entity⟩
$f(\textit{kiss},\text{obj}) = 345$	0.5	⟨Socrates⟩...⟨person⟩⟨life_form⟩⟨CAUSAL_AGENT⟩⟨entity⟩
	0.995	⟨Socrates⟩...⟨PERSON⟩⟨life_form⟩⟨causal_agent⟩⟨entity⟩
(⟨dream⟩, *remember*, obj)	0.0005	⟨dream⟩...⟨preoccupation⟩⟨cognitive_state⟩⟨STATE⟩
	0.05	⟨dream⟩...⟨preoccupation⟩⟨cognitive_state⟩⟨STATE⟩
$f(\textit{remember},\text{obj}) = 1,982$	0.5	⟨dream⟩...⟨preoccupation⟩⟨COGNITIVE_STATE⟩⟨state⟩
	0.995	⟨dream⟩...⟨PREOCCUPATION⟩⟨cognitive_state⟩⟨state⟩
(⟨man⟩, *see*, obj)	0.0005	⟨man⟩...⟨mammal⟩...⟨ANIMAL⟩⟨life_form⟩⟨entity⟩
	0.05	⟨man⟩...⟨MAMMAL⟩...⟨animal⟩⟨life_form⟩⟨entity⟩
$f(\textit{see},\text{obj}) = 16,757$	0.5	⟨man⟩...⟨MAMMAL⟩...⟨animal⟩⟨life_form⟩⟨entity⟩
	0.995	⟨MAN⟩...⟨mammal⟩...⟨animal⟩⟨life_form⟩⟨entity⟩
(⟨belief⟩, *abandon*, obj)	0.0005	⟨belief⟩⟨mental_object⟩⟨cognition⟩⟨PSYCHOLOGICAL_FEATURE⟩
	0.05	⟨belief⟩⟨MENTAL_OBJECT⟩⟨cognition⟩⟨psychological_feature⟩
$f(\textit{abandon},\text{obj}) = 673$	0.5	⟨BELIEF⟩⟨mental_object⟩⟨cognition⟩⟨psychological_feature⟩
	0.995	⟨BELIEF⟩⟨mental_object⟩⟨cognition⟩⟨psychological_feature⟩
(⟨nightmare⟩, *have*, obj)	0.0005	⟨nightmare⟩⟨dreaming⟩⟨IMAGINATION⟩...⟨psych_feature⟩
	0.05	⟨nightmare⟩⟨dreaming⟩⟨IMAGINATION⟩...⟨psych_feature⟩
$f(\textit{have},\text{obj}) = 93,683$	0.5	⟨nightmare⟩⟨DREAMING⟩⟨imagination⟩...⟨psych_feature⟩
	0.995	⟨nightmare⟩⟨DREAMING⟩⟨imagination⟩...⟨psych_feature⟩

Table 3.3: Example levels of generalisation for different values of α; the selected level is shown in upper case

generalisation for strongly selecting verbs, such as *drink* and *eat*, and a greater range of levels for weakly selecting verbs such as *see*. This is because any significant difference in probabilities is likely to be more marked for a strongly selecting verb, and likely to be significant over a wider range of α values. The table only provides anecdotal evidence, but seems to support this argument.

To show how the level of generalisation changes with sample size, we used a fixed α level of 0.05 and the same (c,v,obj) triples as in Table 3.3, but varied the counts in the contingency tables. The results are shown in Table 3.4. The % column gives the amount by which the counts were varied. The 100% row used the same counts as in Table 3.3; the 50% row used these counts multiplied by 0.5; the 10% row used these counts multiplied by 0.1, and so on. The table suggests that the level of generalisation becomes more general as the sample size decreases. This is to be expected, since any difference in probability estimates is less likely to be significant for tables with low counts.

To obtain more of an overall idea of how the level of generalisation varies with changes in α and sample size, we took $3,000$ (c,v,obj) triples and calculated the difference in depth between c and $\texttt{top}(c,v,r)$ for each triple. An average difference in depth was then calculated. The triples were obtained by first extracting $3,000$ (n,v,r) triples from the BNC, where n is a noun, and then using that sense of n that is most probable given v and the object slot. The triples containing

(c,v,r), $f(v,r)$	%	
(⟨coffee⟩, *drink*, obj)	100	⟨coffee⟩⟨BEVERAGE⟩⟨liquid⟩⟨fluid⟩...⟨object⟩⟨entity⟩
	50	⟨coffee⟩⟨BEVERAGE⟩⟨liquid⟩⟨fluid⟩...⟨object⟩⟨entity⟩
$f(\textit{drink}, \text{obj}) = 849$	10	⟨coffee⟩⟨beverage⟩⟨liquid⟩⟨FLUID⟩...⟨object⟩⟨entity⟩
	1	⟨coffee⟩⟨beverage⟩⟨liquid⟩⟨fluid⟩...⟨object⟩⟨entity⟩⟨ROOT⟩
(⟨hotdog⟩, *eat*, obj)	100	⟨hotdog⟩...⟨DISH⟩⟨nourishment⟩⟨food⟩...⟨entity⟩
	50	⟨hotdog⟩...⟨DISH⟩⟨nourishment⟩⟨food⟩...⟨entity⟩
$f(\textit{eat}, \text{obj}) = 1,703$	10	⟨hotdog⟩...⟨dish⟩⟨NOURISHMENT⟩⟨food⟩...⟨entity⟩
	1	⟨hotdog⟩...⟨dish⟩⟨nourishment⟩⟨food⟩...⟨entity⟩⟨ROOT⟩
(⟨Socrates⟩, *kiss*, obj)	100	⟨Socrates⟩...⟨life_form⟩⟨CAUSAL_AGENT⟩⟨entity⟩
	50	⟨Socrates⟩...⟨life_form⟩⟨CAUSAL_AGENT⟩⟨entity⟩
$f(\textit{kiss}, \text{obj}) = 345$	10	⟨Socrates⟩...⟨life_form⟩⟨causal_agent⟩⟨ENTITY⟩
	1	⟨Socrates⟩...⟨life_form⟩⟨causal_agent⟩⟨entity⟩⟨ROOT⟩
(⟨dream⟩, *remember*, obj)	100	⟨dream⟩...⟨preoccupation⟩⟨cognitive_state⟩⟨STATE⟩
	50	⟨dream⟩...⟨preoccupation⟩⟨cognitive_state⟩⟨STATE⟩
$f(\textit{remember}, \text{obj}) = 1,982$	10	⟨dream⟩...⟨preoccupation⟩⟨cognitive_state⟩⟨state⟩⟨ROOT⟩
	1	⟨dream⟩...⟨preoccupation⟩⟨cognitive_state⟩⟨state⟩⟨ROOT⟩
(⟨man⟩, *see*, obj)	100	⟨man⟩...⟨mammal⟩...⟨animal⟩⟨LIFE_FORM⟩⟨entity⟩
	50	⟨man⟩...⟨mammal⟩...⟨animal⟩⟨LIFE_FORM⟩⟨entity⟩
$f(\textit{see}, \text{obj}) = 16,757$	10	⟨man⟩...⟨mammal⟩...⟨animal⟩⟨LIFE_FORM⟩⟨entity⟩
	1	⟨man⟩...⟨mammal⟩...⟨animal⟩⟨life_form⟩⟨entity⟩⟨ROOT⟩
(⟨belief⟩, *abandon*, obj)	100	⟨belief⟩...⟨cognition⟩⟨PSYCHOLOGICAL_FEATURE⟩
	50	⟨belief⟩...⟨cognition⟩⟨PSYCHOLOGICAL_FEATURE⟩
$f(\textit{abandon}, \text{obj}) = 673$	10	⟨belief⟩...⟨cognition⟩⟨psychological_feature⟩⟨ROOT⟩
	1	⟨belief⟩...⟨cognition⟩⟨psychological_feature⟩⟨ROOT⟩
(⟨nightmare⟩, *have*, obj)	100	⟨nightmare⟩...⟨IMAGINATION⟩...⟨process⟩...⟨psych_feature⟩
	50	⟨nightmare⟩...⟨IMAGINATION⟩...⟨process⟩...⟨psych_feature⟩
$f(\textit{have}, \text{obj}) = 93,683$	10	⟨nightmare⟩...⟨imagination⟩...⟨PROCESS⟩...⟨psych_feature⟩
	1	⟨nightmare⟩...⟨imagination⟩...⟨process⟩...⟨PSYCH_FEATURE⟩

Table 3.4: Example levels of generalisation for different quantities of data; the selected level is shown in upper case

the nouns were obtained using the same procedure as that used for the pseudo disambiguation experiments described in Chapters 4 and 6.

To give an example of how the difference in depth was calculated, suppose ⟨dog⟩ generalised to ⟨placental_mammal⟩ via ⟨canine⟩ and ⟨carnivore⟩; in this case the difference would be 3. The results for various levels of α and different sample sizes are shown in Table 3.5, and the % columns are to be interpreted as for Table 3.4. Reading down a column shows how the difference in depth varies with α, and reading across a row shows how the difference varies with sample size. The results demonstrate clearly the trends suggested by the previous two tables.

As a final comparison, the average difference in depth between c and $\texttt{top}(c,v,r)$ was calculated for the same $3,000$ triples, with 100% of the counts, but this time using the Pearson chi-squared statistic. The results comparing X^2 and G^2 are given in Table 3.6. The results show clearly that the average level of generalisation is slightly higher for G^2 than X^2. This can be explained by the fact that G^2 provides a more conservative test than X^2 when counts in the table are low (Agresti 1990). Thus there will be some low count tables for which X^2 returns a significant result, and G^2 returns a non-significant result, thereby forcing the level of generalisation higher when using G^2.

α	100%	50%	10%	1%
0.0005	3.3	3.9	5.0	5.6
0.05	2.8	3.5	4.6	5.6
0.5	2.1	2.9	4.1	5.4
0.995	1.2	1.5	2.6	3.9

Table 3.5: The extent of generalisation for different values of α and sample sizes

α	G^2	X^2
0.0005	3.3	3.0
0.05	2.8	2.5
0.5	2.1	1.9
0.995	1.2	1.2

Table 3.6: The extent of generalisation for G^2 and X^2

3.6 Use of the chi-squared test in corpus-based NLP

In this section the conditions required for the appropriate application of a chi-squared test are considered, followed by discussion of whether to use the G^2 or X^2 statistic. Finally, there is a response to arguments from Kilgarriff (Kilgarriff 1996; Kilgarriff and Rose 1998), who claims that the chi-squared test is not appropriate for use in corpus-based linguistics.

The chi-squared test assumes a random sample where each element of the sample is classified to correspond to one (and only one) of the cells in the contingency table. The sample for Table 3.2, for example, is assumed to have been drawn from the population of liquids appearing in the object position of some verb. Each liquid is then classified according to the set of concepts to which it belongs, and according to whether it is an object of the verb *drink*, or some other verb. These assumptions are not met in practice. One violation is that liquids appearing in the object positions of verbs do not appear independently of each other in a corpus, thus violating the assumption of a random sample. However, the words supplying the concepts in the sample are not contiguous (since they are objects of different verbs) and may be a good distance apart. Thus, in practice, the violation of the independence assumption may not be too bad. Another violation is that, due to the DAG nature of WordNet, some elements of the sample may belong to more than one cell in the table. But since WordNet is a close approximation to a tree (only around 1% of the nodes have more than one parent), this is not expected to be a problem in practice.

Another potential objection is that the chi-squared test is only applicable to contingency tables that have a reasonable number of counts. What ‘reasonable’ means in this context is a moot point, and there is no one rule to cover all cases (Agresti 1990). A rule of thumb often found in text books is that the expected values should be greater than 5 for the sampling distribution of X^2 to be a good approximation to the true chi-squared distribution (Larson 1982). This is sometimes extended to allow some of the expected values to be as low as 1, if the table contains a large number of cells. These guidelines are clearly going to be violated for many of the tables made up of counts from corpora, particularly those corresponding to concepts near the foot of the hierarchy.

One response would be to apply some kind of thresholding, and either ignore any small counts in the table, or only apply the test to tables with large enough counts. Ribas (1995b), Li and Abe (1998), McCarthy (1997) and Wagner (2000) all use some kind of thresholding when dealing with counts in the hierarchy (although not in the context of a chi-squared test). Another response is to use Fisher’s exact test (Agresti 1996). Pedersen (1996) argues for the use of this test in corpus-based linguistics. The application Pedersen considers is the discovery of highly associated bigrams, although the results using Fisher’s exact test are very similar to those obtained using the

G^2 statistic. The advantage of this test is that it can be applied to any contingency table, irrespective of the size of the counts. The main disadvantage is that it is computationally expensive, especially for large contingency tables.

What we have found in practice is that applying the chi-squared test to tables with low counts tends to produce an insignificant result, and the null hypothesis is not rejected. This is especially true for the more conservative G^2 statistic. The consequences of this for the generalisation procedure are that low count tables tend to result in the procedure moving up to the next node in the hierarchy. This behaviour is clearly demonstrated in Tables 3.4 and 3.5. But given that the purpose of the generalisation is to overcome the sparse data problem, this behaviour is desirable, and therefore we do not modify the test for tables with low counts.

The next issue to consider is which statistic to use. Dunning (1993) argues that G^2 is more suitable for corpus-based linguistics than X^2, and Chapter 2 described Dunning's experiment comparing the use of X^2 and G^2 to identify highly associated bigrams. Dunning's claim is that, for small samples, the sampling distribution of G^2 is a better approximation to the chi-squared distribution than the sampling distribution of X^2. However, in Chapter 2 we presented a quotation from Agresti 1996 which contradicts this claim. A more likely explanation lies in the conservative nature of G^2, which means that X^2 is more likely to return a significant result for a table based on small counts. This would explain Dunning's bigram results, in which pairs of words occurring infrequently in the corpus obtain high scores according to X^2 but not G^2.

Note that, for some applications, it may make little difference to the performance whether G^2 or X^2 is used. The results for a PP–attachment task described in Chapter 6 are very similar for both statistics. In fact, the use of X^2 may even lead to better results for some applications. The results of a pseudo disambiguation task, also described in Chapter 6, are at least as good when using X^2, if not better, than when using G^2. The key point seems to be whether the application is one that benefits from a more conservative test (the discovery of highly associated bigrams appears to be such an application) or whether the application benefits from a less conservative test (later results suggest the pseudo disambiguation task described in Chapter 6 is such an application).

The final criticisms relate to the fact that the results of the chi-squared test are highly dependent on the size of the data sample. Kilgarriff argues that the chi-squared test is not appropriate for hypothesis testing in corpus linguistics for this very reason (Kilgarriff 1996; Kilgarriff and Rose 1998). He criticises work by Hofland and Johansson (1982), who use a chi-squared test to try and identify differences in word frequencies in British and American English.

For any given word, Hofland and Johansson frame the following null hypothesis: that the probability of drawing that word at random from the population of British English is the same as drawing that word at random from the population of American English. Hofland and Johansson found significant differences for many words based on counts obtained from corpora, which led them to reject the null hypothesis in each case, and interpret the results as reflecting an underlying difference in word usage between British and American speakers.

Kilgarriff raised doubts about this analysis by taking two samples from the same source (the BNC) and applying the chi-squared test to counts obtained from the two samples. What he found is that even words from these samples resulted in significant differences. Moreover, the differences were often extremely significant, particularly for the more common words. Kilgarriff argues the reason for this is that the null hypothesis is clearly false: words in a corpus are not drawn at random from some larger population because of the dependencies that exist between words. The only question is whether there is enough evidence to indicate the null hypothesis is false, with confidence. Larger samples are more likely to provide that evidence, and hence the outcome of the test is highly dependent on the sample size.

The point of this discussion is not to dispute Kilgarriff's analysis, but we do argue that the dependence of the chi-squared test on the sample size is a *positive* feature for the application we are considering. The reason for this is that the statistical test is required to do two things: one, find a set that is representative of the given concept, and, two, locate areas where there are plenty of counts; and, since the point of this work is to overcome the sparse data problem, the second consideration should override the first. The chi-squared test has this overriding effect built in automatically (particularly when using the conservative G^2 statistic), since it measures the significance of an association, rather than the strength of an association. The test will only return a significant result if there are enough counts to indicate that the observed association is unlikely to have occurred by chance. This property of the test is demonstrated clearly in Tables 3.4 and 3.5.

Wagner (2000) argues that any test used to determine selectional preferences should not exhibit this behaviour. His criticisms are aimed specifically at Li and Abe's use of MDL, as the level of generalisation returned by their system is also heavily dependent on the size of the sample. Wagner's criticism is that large sample sizes tend to lead to over-generalisation, and small sample sizes lead to under-generalisation. One could argue that, if the task is to acquire selectional preferences of the kind Wagner has in mind, this data dependence is undesirable: the generalisation for a kind of food in the object position of *eat* should be around $\langle\texttt{food}\rangle$, whatever the size of the sample. However, since we are interested in probability estimation, rather than the acquisition of selectional preferences (as Wagner describes it), this is not a criticism that can be applied to our approach. We argue that, if the task is to estimate $p(\langle\texttt{beefburger}\rangle|\mathit{eat},\texttt{obj})$, and there already exist lots of data about eating beefburgers, there is no need to generalise to $\langle\texttt{food}\rangle$. Indeed, we show in Chapter 6 that, for some tasks, it can be harmful to generalise unnecessarily.

Chapter 4

Estimating Sense Frequencies from Incomplete Data

In the previous chapter, relative frequency estimates of probabilities were defined in terms of $f(c,v,r)$: the number of times that concept c appears in position r of verb v. If the data used to estimate the probabilities contains WordNet senses, obtaining $f(c,v,r)$ is a trivial counting task. There are data sets that have been tagged with WordNet senses, for example the SemCor data (Miller et al. 1993), but not in the volumes required here. The corpus described by Miller et al. is a manually tagged subset of the Brown corpus, containing only a few hundred thousand words. To obtain enough data to accurately estimate the probabilities of the previous chapter, an automatic method is required, which leaves two options: use an existing disambiguation system or develop a new approach.

The problem with using an existing system is that not all are publically available, and the most successful are complex systems using a combination of knowledge sources that could not easily be recreated. In addition, it is not clear that there has yet been developed an accurate, efficient system that can be applied to large volumes of unrestricted text. The problem of word sense disambiguation (WSD) is a difficult, open problem (Resnik and Yarowsky 1997). Previous approaches to this problem, within the context of acquiring selectional preferences (SPs), have consisted of simple WSD algorithms (Ribas 1995a; McCarthy 1997; Resnik 1997), but they have not been very successful. For these reasons we have developed a novel approach, a feature of which is that it uses the generalisation procedure described in the previous chapter.

Both Resnik (1997) and McCarthy (1997) make the point that SPs and WSD are closely linked, in that acquiring SPs requires sense disambiguated data, and approaches to WSD can make use of SPs. This circularity is exploited here as part of an iterative procedure, in which sense frequencies for the current iteration are estimated using 'selectional preference scores' derived from the previous iteration. The scores are used to compare the alternative senses for a noun in the data, so that senses with higher scores receive more of the count. The first step in the re-estimation is to simply split the count for a noun evenly among the noun's senses. Resnik (1998) suggests using this as a first step in a re-estimation process, although he does not suggest how the re-estimation might proceed. Before doing so, we first explain how the apparently crude technique of splitting the count equally can work at all.

4.1 The first step in the re-estimation procedure

In the previous chapter, the approach to estimating sense frequencies was to split the count for a noun n appearing in position r of verb v equally among the senses of the noun:

$$\hat{f}(c,v,r) = \sum_{n \in \mathsf{syn}(c)} \frac{f(n,v,r)}{|\mathsf{cn}(n)|} \tag{4.1}$$

This may appear to be a crude solution to the problem of ambiguous data, but, in practice, it works surprisingly well. The reason is that counts for sets of concepts tend to accumulate in the right places. To see why, consider this example adapted from Resnik 1998. (Resnik notes that a similar point is made by Yarowsky (1992).) Consider estimating probabilities for the object position of the verb *drink*, and suppose that *drink wine* and *drink water* occur as part of the data. The word *water* is a member of seven senses in WordNet, and *wine* is a member of two senses. Thus, for these data items, splitting the count equally leads to each sense of *water* receiving 0.14 counts and each sense of *wine* 0.5 counts. But note that, with regard to *sets* of concepts, only those sets containing senses of both *wine* and *water*, such as ⟨`beverage`⟩, will accumulate counts. The counts for the incorrect senses will be randomly dispersed throughout the hierarchy as noise, and areas where counts would be expected to accumulate, such as under ⟨`beverage`⟩ in this example, will receive the majority of the overall count. As will be shown later, this accumulation effect means that performance in applications can be good, even if this simple estimation technique is used.

However, there is an obvious problem with this approach: although counts for sets tend to accumulate in the right places, counts can be greatly underestimated. In the previous example, $\hat{f}(\langle\texttt{beverage}\rangle, \mathit{drink}, \mathrm{obj})$ is incremented by only 0.64 counts from the two data instances, rather than the correct value of 2. In addition, as Resnik himself notes, the accumulation process has less effect on sets of concepts low down in the hierarchy, since here the counts have had less chance to accumulate. The example Resnik gives is for *blow nose*. In this case, counts would be expected to be higher for the set dominated by the bodily sense of *nose*, rather than the aircraft sense. However, since both senses are low down in the hierarchy, splitting counts equally is likely to lead to a similar count for each set. For the same reason, counts for individual concepts, as opposed to sets of concepts, are likely to be inaccurate.

In response to this, we note that the accumulation of counts leads to an obvious strategy: use the fact that correct senses are likely to be members of sets where counts have accumulated as a way of re-distributing the count. Continuing with the *drink wine* example, *wine* has a beverage sense and a colour sense in WordNet. If the above strategy is used, equal counts will be given to each sense on the first iteration, but, on subsequent iterations, more of the count will be given to the beverage sense. This is because counts would accumulate under ⟨`beverage`⟩ for the object position of *drink* and not under ⟨`colour`⟩.

One issue to consider is how to determine a representative set for a concept. We have been assuming that ⟨`beverage`⟩ and ⟨`colour`⟩ are suitable for the two senses of *wine*, but a procedure is needed which determines this automatically. The procedure needs to find a hypernym for each alternative sense, such that the hypernym is high enough for counts to have accumulated in the set dominated by the hypernym; however, it should not be so high that the alternative senses cannot be distinguished. An example of a hypernym that is too high is ⟨`root`⟩, the notional root of the hierarchy, since if ⟨`root`⟩ were chosen for both senses of wine, there would be no way to distinguish between the senses. Another reason not to go too high is that the sets need to be, in some sense, representative of the senses. Suppose *eat chip* occurs in the data, and the food sense of *chip* and the electronic sense need to be distinguished. It would not be appropriate to represent the electronic sense using ⟨`entity`⟩, since this would not capture the fact that this sense is strongly negatively associated with the object position of *eat*. A more suitable hypernym would be something like ⟨`artifact`⟩.

If the accumulation of counts is interpreted in terms of the association between verbs and sets of concepts, then the procedure for selecting a hypernym described in the previous chapter can be used to determine a representative set. The reasons for this are given in Section 4.3. In the next section we show how such associations can be used to split the count, and also give the details of the re-estimation algorithm.

4.2 Using a measure of association to re-distribute the count

An alternative way to think about the accumulation effect is that counts accumulate for sets that are positively associated with the argument position of the verb. The set $\overline{\langle\texttt{beverage}\rangle}$ is more associated with the object position of the verb *drink* than $\overline{\langle\texttt{colour}\rangle}$, which explains why counts accumulate more under $\langle\texttt{beverage}\rangle$ than $\langle\texttt{colour}\rangle$. Hence, what is needed is a measure of association that can be used to give more of the count to senses belonging to positively associated sets. The measure used here is the 'association norm,' taken from Abe and Li 1996.

The association norm, $\mathrm{A}(C,v,r)$, is a measure of association between a verb, v, and set of concepts, C, assuming an argument position, r:[1]

$$\mathrm{A}(C,v,r) = \frac{p(C,v|r)}{p(C|r)p(v|r)} \tag{4.2}$$

A similar score was originally proposed by Church and Hanks (1990) for use in corpus-based NLP, based on the information theoretic notion of mutual information, and there have been other similar proposals such as Resnik's selectional association. The possible values for $\mathrm{A}(C,v,r)$ range between zero and positive infinity; a value between zero and one indicates a negative association, and a value greater than one indicates a positive association.

The point of the association norm is to compare the joint probability of observing v with some member of class C, $p(C,v|r)$, with the probability of this observation if C and v were independent, $p(C|r)p(v|r)$. If C and v are highly associated, we would expect $p(C,v|r)$ to be much greater than $p(C|r)p(v|r)$, resulting in a high value for the association norm. An example of a verb and class that are highly associated is the verb *eat* and the class $\overline{\langle\texttt{food}\rangle}$ (assuming the object position). These would also be expected to have a high association norm. An easy way to see this is to note that $\mathrm{A}(C,v,r)$ can be written as follows:

$$\mathrm{A}(C,v,r) = p(C|v,r)/p(C|r) \tag{4.3}$$

We would expect $p(\overline{\langle\texttt{food}\rangle}|\mathit{eat},\mathrm{obj})$ to be much greater than $p(\overline{\langle\texttt{food}\rangle}|\mathrm{obj})$, since the probability of finding an item of food in the object position of *eat* is much greater than the probability of finding an item of food in the object position of any verb. This results in a high value for $\mathrm{A}(\overline{\langle\texttt{food}\rangle},\mathrm{eat},\mathrm{obj})$.

An estimate for $\mathrm{A}(C,v,r)$ can be obtained by using relative frequency estimates of the relevant probabilities:

$$\hat{\mathrm{A}}(C,v,r) = \frac{\hat{p}(C|v,r)}{\hat{p}(C|r)} \tag{4.4}$$

Such estimates can be used to split the count for a noun among its senses. The obvious strategy is to give sense c of noun n the following proportion of the count, where $[c,v,r]$ denotes the representative set for c (in position r of verb v):

$$\frac{\hat{\mathrm{A}}([c,v,r],v,r)}{\sum_{c'\in\mathsf{cn}(n)}\hat{\mathrm{A}}([c',v,r],v,r)} \tag{4.5}$$

The count is split according to the ratio of the association norm for $[c,v,r]$ relative to the total association norm summed over the senses of the noun.

To give an example, suppose that the sets representing the two senses of *wine*, relative to the object position of *drink*, are $\overline{\langle\texttt{beverage}\rangle}$ and $\overline{\langle\texttt{colour}\rangle}$, and that the estimated association norms are 147 and 46 respectively. These values were calculated from data taken from a subset of the BNC, obtained using the system of Briscoe and Carroll (1997). The calculation for $\overline{\langle\texttt{beverage}\rangle}$ is given

[1] The association norm is defined slightly differently by Abe and Li, since they are concerned with the association between a verb and a *noun*, rather than a verb and a set of concepts.

$$\begin{aligned}
\hat{p}(\overline{\langle\texttt{beverage}\rangle}|\textit{drink},\mathrm{obj}) &= \frac{\hat{f}(\overline{\langle\texttt{beverage}\rangle},\textit{drink},\mathrm{obj})}{\hat{f}(\textit{drink},\mathrm{obj})}\\
&= 261.0/1,024\\
&= 0.255
\end{aligned}$$

$$\begin{aligned}
\hat{p}(\overline{\langle\texttt{beverage}\rangle}|\mathrm{obj}) &= \frac{\hat{f}(\overline{\langle\texttt{beverage}\rangle},\mathrm{obj})}{\hat{f}(\mathrm{obj})}\\
&= 2,628.7/1,508,950\\
&= 0.00174
\end{aligned}$$

$$\begin{aligned}
\hat{\mathrm{A}}(\overline{\langle\texttt{beverage}\rangle},\textit{drink},\mathrm{obj}) &= 0.255/0.00174\\
&= 147
\end{aligned}$$

Figure 4.1: Calculation of $\hat{\mathrm{A}}(\overline{\langle\texttt{beverage}\rangle},\textit{drink},\mathrm{obj})$, based on data taken from the BNC

in Figure 4.1 In this case, around three quarters of the count would be given to the beverage sense: $147/(147+46) = 0.76$. In fact, the estimated association norm for $\overline{\langle\texttt{colour}\rangle}$ is too high. Later, in Figure 4.4, the value for $\hat{\mathrm{A}}(\overline{\langle\texttt{food}\rangle},\textit{eat},\mathrm{obj})$ is calculated as 31, which suggests that the estimate for $\overline{\langle\texttt{colour}\rangle}$ is being affected by noise. (The reasons for this over-estimation are explained in Section 4.4.) A more suitable hypernym would be the higher concept $\langle\texttt{abstraction}\rangle$; for the set $\overline{\langle\texttt{abstraction}\rangle}$, the estimated association norm is only 0.9. Using this set to represent the colour sense would mean that almost all of the count would go to the beverage sense: $147/(147+0.9) = 0.99$.

We can now give the details of the re-estimation algorithm. The first estimate of $f(c,v,r)$ is obtained by splitting the count for any noun equally among its senses:

$$\hat{f}^0(c,v,r) = \sum_{n\in\mathsf{syn}(c)} \frac{f(n,v,r)}{|\mathsf{cn}(n)|} \tag{4.6}$$

Subsequent estimates are obtained by re-distributing the count according to the relevant association scores, where the scores are estimated using counts from the previous iteration:

$$\hat{f}^{m+1}(c,v,r) = \sum_{n\in\mathsf{syn}(c)} f(n,v,r)\frac{\hat{\mathrm{A}}^m([c,v,r],v,r)}{\sum_{c'\in\mathsf{cn}(n)}\hat{\mathrm{A}}^m([c',v,r],v,r)} \tag{4.7}$$

The sum is over nouns in $\mathsf{syn}(c)$ because these are the nouns that can lead to a count for c. The formulae for the relevant estimates are given in Figure 4.2. In the next section, we show how the generalisation procedure given in the previous chapter can be used to determine $[c,v,r]$.

4.3 Determining representative sets

In the previous section, it was shown how $\overline{\langle\texttt{abstraction}\rangle}$ is a better representative of the colour sense of *wine* than $\overline{\langle\texttt{colour}\rangle}$ (at least for the data set in question and the object position of *drink*). The concept $\langle\texttt{abstraction}\rangle$ is a root of one of the nine complete sub-hierarchies in WordNet (see Figure 3.1), which raises the question of why $\overline{\langle\texttt{entity}\rangle}$, which is also one of the nine complete sub-hierarchies, is not a good representative of the beverage sense. (The estimated association norm for $\overline{\langle\texttt{entity}\rangle}$ in the object position of *drink* is only 1.7, hardly representative of the fact that the beverage sense of *wine* is highly associated with *drink*.) The key observation is that the set

$$\hat{\mathrm{A}}^m(C,v,r) = \frac{\hat{p}^m(C|v,r)}{\hat{p}^m(C|r)}$$

$$\hat{p}^m(C|v,r) = \frac{\hat{f}^m(C,v,r)}{\hat{f}(v,r)}$$

$$\hat{p}^m(C|r) = \frac{\sum_{v\in\mathcal{V}}\hat{f}^m(C,v,r)}{\sum_{v\in\mathcal{V}}\hat{f}(v,r)}$$

$$\hat{f}^m(C,v,r) = \sum_{c\in C}\hat{f}^m(c,v,r)$$

Figure 4.2: Estimates for calculating $\hat{\mathrm{A}}^m(C,v,r)$ for a set of concepts C; $\mathcal{V}$ is the set of verbs in the data

$\overline{\langle\texttt{entity}\rangle}$ is not homogeneous with respect to the object position of *drink*: some entities are drunk, some are not. In contrast, the set $\overline{\langle\texttt{abstraction}\rangle}$ is fairly homogeneous in that, on the whole, kinds of abstraction are rarely drunk.

The set $\overline{\langle\texttt{beverage}\rangle}$ is also homogeneous, which is a suitable representative for the beverage sense. Note that the two sets $\overline{\langle\texttt{abstraction}\rangle}$ and $\overline{\langle\texttt{beverage}\rangle}$ are also 'maximally homogeneous,' in that the sets dominated by the parents of $\langle\texttt{beverage}\rangle$ and $\langle\texttt{abstraction}\rangle$, $\overline{\langle\texttt{liquid}\rangle}$ and $\overline{\langle\texttt{root}\rangle}$ respectively, are not themselves homogeneous. This motivates the idea that we should be looking for maximally homogeneous sets, maximal because we want to allow counts to accumulate and noise to be dispersed. The problem with using $\overline{\langle\texttt{colour}\rangle}$ as a representative of $\langle\texttt{wine}\rangle$ is that $\langle\texttt{colour}\rangle$ is not high enough for this dispersal to have occurred.

One way to recognise that $\overline{\langle\texttt{liquid}\rangle}$ is not homogeneous is to note that the sets dominated by the daughters of $\langle\texttt{liquid}\rangle$ are associated to differing degrees with *drink*. Some liquids are drunk, such as beverages, liquor and water, but some are not, such as ammonia, antifreeze and sheep dip. This motivates a test for homogeneity that compares the levels of association of the daughter sets.[2] So to determine if $\overline{\langle\texttt{liquid}\rangle}$ is a homogeneous set, the values of $\hat{\mathrm{A}}(\overline{\langle\texttt{beverage}\rangle}, \textit{drink}, \mathrm{obj})$, $\hat{\mathrm{A}}(\overline{\langle\texttt{liquor}\rangle}, \textit{drink}, \mathrm{obj})$, $\hat{\mathrm{A}}(\overline{\langle\texttt{ammonia}\rangle}, \textit{drink}, \mathrm{obj})$ and so on, can be compared. If the values are very different, then this suggests that the set $\overline{\langle\texttt{liquid}\rangle}$ is not homogeneous with respect to the object position of *drink*.

Note that this is reminiscent of the procedure described in the previous chapter, where probabilities conditioned on daughter sets were compared. The probabilities being compared were $p(v|\overline{c'_i},r)$, where the $\overline{c'_i}$ are the daughter sets; but note that, since $p(v|r)$ is a constant across the daughter sets, this is equivalent to comparing association norms:

$$\begin{aligned} p(v|\overline{c'_i},r) &= \frac{p(\overline{c'_i},v|r)}{p(\overline{c'_i}|r)} && (4.8)\\ &= \frac{p(\overline{c'_i},v|r)}{p(\overline{c'_i}|r)p(v|r)}p(v|r) && (4.9)\\ &= \mathrm{A}(\overline{c'_i},v,r)p(v|r) && (4.10) \end{aligned}$$

In addition, the procedure finds maximally homogeneous sets, in that it only stops moving up the hierarchy when a concept is found whose daughter sets are associated to differing degrees with

[2] The term 'daughter set' is used to denote the set of concepts dominated by a daughter.

the verb. Thus it appears that the procedure can be applied directly to the problem of determining $[c, v, r]$.

However, there are some differences between the problems being addressed in this and the previous chapter. In the previous chapter the problem was to find a generalisation level that would lead to a reasonable probability estimate. In this chapter the problem is to find a level where counts have accumulated and the noise dispersed sufficiently. A solution to both problems lies in finding homogeneous sets; the difference lies in the *degree* of homogeneity that is likely to be optimal in each case. For the probability estimation problem, it may be that the difference in association norms needs to be relatively small for a class-based probability estimate to be a useful estimate. Results presented in Chapter 6 suggest that, for some disambiguation tasks, this is indeed the case. Another way to think of this is that, for some tasks, the optimal level of generalisation is quite low in the hierarchy, on the whole. In contrast, the re-estimation problem is likely to favour a level of generalisation that is quite high, on the whole, since it is here that counts have accumulated and noise dispersed.

Despite these differences, the procedure can be adapted to both problems. The degree of homogeneity required can be controlled by the parameter α, the level of significance of the chi-squared test. The value of α controls the overall level of generalisation: a high value for α results in a low level of generalisation, on the whole, and a low value for α results in a high level of generalisation. Results from the previous chapter clearly demonstrate this. One way to set a value for α would be to estimate counts using a range of α values, and use a held-out test set to choose those counts that give the best performance on the task in hand.

Another useful feature of the procedure, within the context of the re-estimation problem, is that it employs a significance test to find homogeneous sets. This implies that the procedure automatically finds areas where counts have accumulated, since it is only here that there will be enough data to return a significant result for the chi-squared test. This point is especially true when the more conservative G^2 statistic is used and a low value for α.

As a final comment, a point of clarification is needed. The previous chapter showed that the chosen level of generalisation is dependent on the size of the data sample, as well as on the value of α. Thus the notion of homogeneity being used here is not an absolute notion, but a relative one, relative to the sample. If the procedure determines a maximally homogeneous set that does not accord with intuition, this should not be automatically considered a failure. A comment in Clark and Weir 1999 states that $\langle\texttt{food}\rangle$ is heterogeneous with respect to the object position of *eat*. The reason given is that $\langle\texttt{beverage}\rangle$ is classified as a kind of food (as well as a liquid), and beverages are not eaten, on the whole. We now view this analysis as mistaken, since there is no one right answer to the question of which sets are homogeneous, and the argument for finding homogeneous sets is intended primarily to give the intuition behind the approach; the 'correct' generalisation level is ultimately the one that leads to the most accurate estimates or, alternatively, the best performance in a given application.

4.4 Criticisms of the association norm

The use of the association norm and similar measures has been criticised in the literature. Dunning (1993) argues that estimates are prone to over-estimation when based on small counts. The following example, based on data taken from a subset of the BNC, shows how over-estimation can occur. The example is for the set of concepts $\langle\texttt{twelve}\rangle$ and the object position of the verb *cajole*. The concept $\langle\texttt{twelve}\rangle$ has the synset $\{$*twelve*, *12*, *XII*, *dozen*$\}$, and has one child with the synset $\{$*boxcars*$\}$.[3] Members of the set $\langle\texttt{twelve}\rangle$ occurred once in the object position of *cajole*, and the verb occurred five times in total with an object. The calculation for $\hat{A}(\langle\texttt{twelve}\rangle, \mathit{cajole}, \mathrm{obj})$, based on these counts, is given in Figure 4.3, and results in a value of 687. As a comparison, consider the value for $\hat{A}(\langle\texttt{food}\rangle, \mathit{eat}, \mathrm{obj})$, based on the same data. The calculation for this is given in Figure 4.4,

[3] The term *boxcars* apparently refers to the two sixes that can be thrown in a game of dice.

$$\begin{aligned}
\hat{p}(\overline{\langle\texttt{twelve}\rangle}|\mathit{cajole},\mathrm{obj}) &= \frac{\hat{f}(\overline{\langle\texttt{twelve}\rangle},\mathit{cajole},\mathrm{obj})}{\hat{f}(\mathit{cajole},\mathrm{obj})} \\
&= 1/5 \\
&= 0.200
\end{aligned}$$

$$\begin{aligned}
\hat{p}(\overline{\langle\texttt{twelve}\rangle}|\mathrm{obj}) &= \frac{\hat{f}(\overline{\langle\texttt{twelve}\rangle},\mathrm{obj})}{\hat{f}(\mathrm{obj})} \\
&= 439/1{,}508{,}950 \\
&= 0.000291
\end{aligned}$$

$$\begin{aligned}
\hat{\mathrm{A}}(\overline{\langle\texttt{twelve}\rangle},\mathit{cajole},\mathrm{obj}) &= 0.200/0.000291 \\
&= 687
\end{aligned}$$

Figure 4.3: Calculation of $\hat{\mathrm{A}}(\overline{\langle\texttt{twelve}\rangle},\mathit{cajole},\mathrm{obj})$

resulting in a value of 31. Clearly, the value for $\hat{\mathrm{A}}(\overline{\langle\texttt{twelve}\rangle},\mathit{cajole},\mathrm{obj})$ is greatly over-estimated, since $\overline{\langle\texttt{twelve}\rangle}$ and *cajole* are not particularly associated, and yet their score is much higher than the score for the highly associated $\overline{\langle\texttt{food}\rangle}$ and *eat*. The problem is that the estimate of 0.2 for $p(\overline{\langle\texttt{twelve}\rangle}|\mathit{cajole},\mathrm{obj})$ is much too high, and arises because one of the few instances of *cajole* happens to have a member of $\overline{\langle\texttt{twelve}\rangle}$ as an object.

This possibility of over-estimation arises because the association norm is a measure of association, rather than a measure of significance. The score takes no account of the fact that any co-occurrence could be entirely due to chance, a possibility that is more likely when only a small amount of data is being considered. However, the score can be effective if used appropriately. The first point to note is that over-estimation is unlikely to be a problem for common verbs. This is because $\hat{p}(C|v,r)$ is unlikely to be greatly over-estimated, since many occurrences of v are being considered.

The score can also be used with uncommon verbs, but only with large sets. To see this, first note that $\hat{\mathrm{A}}(C,v,r)$ can be written as $\hat{p}(v|C,r)/\hat{p}(v|r)$:

$$\begin{aligned}
\hat{\mathrm{A}}(C,v,r) &= \frac{\hat{p}(C|v,r)}{\hat{p}(C|r)} && (4.11) \\
&= \frac{f(C,v,r)}{f(v,r)} / \frac{f(C,r)}{f(r)} && (4.12) \\
&= \frac{f(C,v,r)}{f(C,r)} / \frac{f(v,r)}{f(r)} && (4.13) \\
&= \frac{\hat{p}(v|C,r)}{\hat{p}(v|r)} && (4.14)
\end{aligned}$$

For uncommon verbs, the value of $\hat{p}(v|r)$ will be very small, and so the danger lies in the possibility of a large value for $\hat{p}(v|C,r)$. This can occur when $f(C,r)$ is small, which in turn tends to occur with sets dominated by a concept low down in the hierarchy. However, if $f(C,r)$ is large, then the danger of obtaining too large a value for $\hat{p}(v|C,r)$ is greatly reduced. To help support this idea, note that $\hat{\mathrm{A}}(\overline{\langle\texttt{root}\rangle},v,r) = 1$ for all v and r, irrespective of the number of times v occurs in the data. The value of $\hat{f}(\overline{\langle\texttt{root}\rangle},r)$ is as large as for any set, and it is simply not possible to over-estimate $\hat{\mathrm{A}}(\overline{\langle\texttt{root}\rangle},v,r)$. To give another example, the value for $\hat{\mathrm{A}}(\overline{\langle\texttt{entity}\rangle},\mathit{cajole},\mathrm{obj})$, based on the BNC data, is 0.98. Given this value is less than one, this is unlikely to be an over-estimation, or at least unlikely to be a serious over-estimation. The reason is that $\overline{\langle\texttt{entity}\rangle}$ is a large set, and leads to a

$$\begin{aligned}\hat{p}(\overline{\langle\texttt{food}\rangle}|eat,\text{obj}) &= \frac{\hat{f}(\overline{\langle\texttt{food}\rangle},eat,\text{obj})}{\hat{f}(eat,\text{obj})}\\ &= 711/2,045\\ &= 0.348\end{aligned}$$

$$\begin{aligned}\hat{p}(\overline{\langle\texttt{food}\rangle}|\text{obj}) &= \frac{\hat{f}(\overline{\langle\texttt{food}\rangle},\text{obj})}{\hat{f}(\text{obj})}\\ &= 16,880/1,508,950\\ &= 0.0112\end{aligned}$$

$$\begin{aligned}\hat{\text{A}}(\overline{\langle\texttt{food}\rangle},eat,\text{obj}) &= 0.348/0.0112\\ &= 31\end{aligned}$$

Figure 4.4: Calculation of $\hat{\text{A}}(\overline{\langle\texttt{food}\rangle},eat,\text{obj})$

high value for $\hat{f}(\overline{\langle\texttt{entity}\rangle},r)$, and so $\hat{p}(\text{cajole}|\overline{\langle\texttt{entity}\rangle},\text{obj})$ is not over-estimated.

The conclusion is that, if the association norm is to be applied appropriately, it should be applied to frequent verbs or to sets for which $f(C,r)$ is reasonably high; however, since the re-estimation procedure relies on using sets where plenty of counts have accumulated, this should not be a problem.

4.5 Evaluation

There are two evaluations in this section.[4] The first shows how the estimated counts change as the re-estimation proceeds, and uses some verbs for which the correct count can be inferred. The problem with this evaluation is that it only considers a small number of hand-selected cases. The second evaluation uses the re-estimated counts as part of a pseudo disambiguation task, and compares the results with those obtained using counts from only the first step of the re-estimation algorithm. This is a more comprehensive evaluation, since it uses a wide selection of randomly chosen verbs.

4.5.1 How the counts change over the iterations

A number of (c,v,obj) triples were hand chosen and counts estimated using the re-estimation procedure. The data were again obtained using the system of Briscoe and Carroll, and were taken from around two million words of the BNC. For the purposes of this evaluation, the G^2 statistic was used for the chi-squared test, with an α value of 0.05.

Table 4.1 shows, for various sets of concepts in the object position of a selection of verbs, how the estimated frequencies changed during the re-estimation process. The first column gives the estimates using the technique of splitting the count for a noun equally among its senses. The other columns give the estimates from subsequent iterations of the re-estimation. The estimates appear to be converging after around 10 iterations, although there appears to be very little change after 5 iterations. The final column gives an upper bound on the re-estimated frequencies. It shows how many nouns there are in the data appearing in the object position of the given verb that could possibly be denoting a concept in $\overline{c}$. For example, 95 is the number of times that a noun that could possibly be denoting an item of food appeared in the object position of *eat*.

[4] The first evaluation has been published as part of Clark and Weir 1999, and the second is a new task-based evaluation that is not described in the paper.

$(v,\bar{c})$	$\hat{f}^m(\bar{c},v,\text{obj})$				Limit
	$m=0$	$m=1$	$m=5$	$m=10$	
$(\textit{eat},\overline{\langle\texttt{food}\rangle})$	60.8	85.0	89.6	89.8	95
$(\textit{drink},\overline{\langle\texttt{beverage}\rangle})$	10.5	22.7	23.5	23.4	26
$(\textit{eat},\overline{\langle\texttt{location}\rangle})$	2.0	1.2	1.1	1.1	6
$(\textit{see},\overline{\langle\texttt{object}\rangle})$	237.1	235.7	240.2	240.3	568
$(\textit{hear},\overline{\langle\texttt{person}\rangle})$	90.8	85.5	85.5	85.5	130
$(\textit{enjoy},\overline{\langle\texttt{amusement}\rangle})$	2.9	3.1	3.3	3.3	5
$(\textit{measure},\overline{\langle\texttt{abstraction}\rangle})$	19.1	21.7	23.3	23.4	31

Table 4.1: Examples of re-estimated frequencies

The figures for *eat* and *drink* in the first two rows suggest that the initial estimates can be far too low. Since *eat* and *drink* select so strongly for their objects, we would expect the true frequency to be quite close to the 'Limit' value in the final column. In other words, if there is a noun in the object position of *eat* that could be denoting an item of food, it probably is denoting an item of food. A similar argument applies to the object position of *drink*. Note that, for these cases, the re-estimates do converge quite closely to the limit value, and they increase considerably from the initial estimate. (For completeness, the limit value has been given for all the verbs, although it is of less consequence for weakly selecting verbs.) Another problem with the initial estimates is that they can be too large. This can occur when the argument *violates* the preferences of the verb. The example in the table is for members of $\overline{\langle\texttt{location}\rangle}$ in the object position of *eat*. Note that the re-estimated value has decreased by almost one half from the initial estimate.

The estimates for the weakly selecting verbs do not change as much as for the strongly selecting verbs. The greatest changes, for the verbs in the table, occur for *eat* and *drink*. This is to be expected, since, for weakly selecting verbs, counts from the first step of the re-estimation process will not accumulate in particular areas of the hierarchy. The counts will be spread fairly evenly, and the differences in association norms for alternative senses are likely to be small, which means that, on subsequent iterations, each alternative sense will continue to receive around the same proportion of the count.

The contrast between the figures for different verbs raises the question of how much impact, overall, the re-estimation process is likely to be having. It appears that the re-estimation has a large impact on strongly selecting verbs, but if these make up only a small proportion of the verb population, the overall impact may be minimal. To test this, we estimated, for each $(n,v,\texttt{obj})$ triple in the data, how the distribution of the count had changed over the re-estimation. As a measure of how the distribution over the alternative senses had changed, we used the percentage increase of the count going to the most favoured sense. For example, if, after 10 iterations, 0.99 of the count for *drink wine* went to the favoured, beverage sense, and 0.01 to the remaining, colour sense, the percentage increase would be:

$$(0.99/0.5)-1=98\%$$

The results shown in Table 4.2 are for triples containing nouns with more than one sense; these nouns made up 83% of the data. The results indicate that, for 43% of these triples, the re-estimation is having little effect, but, for 23%, the proportion of the count going to the most favoured sense is at least doubled.

A final point is that the effectiveness of the re-estimation will, to some extent, depend on the size of the data sample. If the number of occurrences of a verb and argument position is small, then WordNet will be sparsely populated with counts in this case, and there will be no way to distinguish between the alternative senses. For very sparsely populated instances of WordNet, the

Percentage Increase	Proportion of Data
0–10	43%
10–50	18%
50–100	16%
100-	23%

Table 4.2: How the distribution of the count changes

generalisation procedure is likely to return the root of the hierarchy for any given sense, in which case the count for a noun would continue to be divided equally among the alternative senses.

4.5.2 A task-based evaluation

The task used to evaluate the re-estimation procedure further is a pseudo disambiguation task similar to that performed by Pereira et al. (1993), and Rooth et al. (1999), and is used again in Chapter 6. The task is an appropriate evaluation because it only uses probabilities of the form $p(c|v,r)$. Any improvement in the frequency estimates will lead to an improvement in the probability estimates, and this should be reflected in the task performance.

The task is to decide which of two verbs, v and v', is more likely to take a given noun, n, as an argument. The test and training data were obtained as follows. A number of verb direct object pairs were extracted from a subset of the BNC, using the system of Briscoe and Carroll. All those pairs containing a noun not in WordNet were removed, and each verb and argument was lemmatised. This resulted in a data set of around $260,000$ (v,n) pairs.

To form a test set, $3,000$ of these pairs were randomly selected, such that each selected pair contained a fairly frequent verb. (Only those verbs that occurred between 100 and $1,000$ times in the data were considered.)[5] Each instance of a selected pair was then deleted from the data. This was to ensure that the test data were unseen. The remaining pairs formed the training data. To complete the test set, a further fairly frequent verb, v', was randomly chosen for each (v,n) pair. The random choice was made according to the verb's frequency in the original data set, subject to the condition that the pair (v',n) did not occur in the training data.

Given the set of (v,n,v') triples, the task is to decide whether (v,n) or (v',n) is the correct pair. Note that the sampling procedure does not guarantee that the correct pair, (v,n), is more plausible than the corresponding incorrect pair, (v',n), since a highly plausible incorrect pair could be generated by chance. (And the parser will produce some erroneous data.) The assumption is that this will occur infrequently in practice.

The decision for each (v,n,v') test triple was made as follows. The probabilities $\hat{p}(c|v,\text{obj})$ and $\hat{p}(c'|v',\text{obj})$ were compared, where c is the concept that maximises $\hat{p}(c''|v,\text{obj})$, and c' is the concept that maximises $\hat{p}(c''|v',\text{obj})$, for $c'' \in \mathsf{cn}(n)$. In other words, the concept was chosen that maximised the relevant probability estimate. The verb noun pair with the highest probability was then chosen as the correct pair. The probability estimates were obtained using the technique described in the previous chapter, and the G^2 statistic was used for the chi-squared test.

Before describing the results, a potential source of confusion needs to be addressed. There are two separate applications of the chi-squared test here, using two potentially different values for α. The first application forms part of the re-estimation procedure, and we refer to the corresponding α value as '*re-est*-α'. The second application forms part of the estimation of the probabilities being compared to make the disambiguation decision. We refer to this α value as '*prob-est*-α'. The optimum value for the two cases of α could be very different.

The results are given in in Table 4.3, for a range of values of *re-est*-α and *prob-est*-α. The first row shows the results obtained when no re-estimation was applied to the counts. The re-estimated

[5] In Chapter 6, a larger training set is used, but a smaller set was used here to ease the computational burden.

prob-est-α *re-est*-α	0.0005	0.05	0.3	0.75	0.995
No re-estimation	66.4	68.1	69.8	72.1	71.8
0.0005	65.6	67.4	69.5	72.2	71.9
0.05	68.8	69.0	70.4	72.2	72.1
0.1	70.0	70.6	70.6	72.3	71.9
0.3	70.0	70.2	70.1	71.3	71.3
0.75	68.8	71.0	70.6	70.8	69.6

Table 4.3: Disambiguation results across a range of α values

counts were obtained from 5 iterations of the re-estimation procedure, since table 4.1 indicated that the re-estimated counts change very little after 5 iterations. The figures in the table are the percentage of correct decisions for the $3,000$ test cases.

The results show that, for low values of *prob-est*-α, the re-estimated counts can improve the performance, but for high values of *prob-est*-α, the re-estimated counts have little impact. The results suggest that the optimum value for *re-est*-α is around 0.1, and for *prob-est*-α around 0.75. (Of course, in practice, these values would need to be optimised on a held-out set, rather then the test set itself.)

The conclusion is that the use of selectional preferences, alone, is not enough for highly accurate WSD. As Resnik (1997) notes, many verbs do not select strongly enough for their arguments for the correct sense to be distinguished. This conclusion is supported by the results in Table 4.2, where around half of the data items were largely unaffected by the re-estimation procedure. A similar conclusion is arrived at in Carroll and McCarthy 2000.

A feature of this chapter is how the procedure for determining a suitable level of generalisation can be applied directly to the problem of finding homogeneous sets. This has provided an additional interpretation of that procedure. We have also shown how different tasks may require different values of α; the optimum value for the re-estimation task is lower than that for the probability estimation task. As will be argued in Chapter 6, this flexibility allowed by the α parameter is a positive feature of the generalisation procedure.

Chapter 5

Integrating the Estimation Techniques into a Parse Selection System

The primary aim of this chapter is to show how the estimation techniques described in Chapter 3 can be integrated into a parse selection system. Parse selection is an obvious application of these techniques, since the importance of lexical information for parse selection is well established, and we saw in Chapter 1 how lexical sense preferences are very similar to lexical dependencies. In addition, previous work using WordNet has only looked at particular structural ambiguities in isolation, and an obvious way to extend this work is to use WordNet for the more general problem of parse selection.

Hindle and Rooth (1991, 1993) were among the first to demonstrate the importance of lexical dependencies for structural disambiguation (and hence parse selection). Their work focused on PP-attachment ambiguity, but other work has shown how lexical dependencies can be used to resolve other ambiguities, such as relative clause attachment, noun-noun compound, and coordination ambiguities (Fisher and Riloff 1992; Lauer 1995; Resnik 1999b). These ideas have been extended to statistical parsing, and many of the most cited statistical parsers incorporate lexical dependencies in some form (Jelinek et al. 1994; Magerman 1995; Collins 1996, 1997; Eisner 1996b; Goodman 1997; Bod 1998; Ratnaparkhi 1999; Charniak 1997, 2000).[1]

The work in this chapter is motivated by that of Collins (1997), who uses a top-down stochastic process to generate phrase structure parse trees, together with a history-based model (Black et al. 1993) to define the probability of a parse. A history-based model is simply a sequence of decisions that generates a structure in some canonical order (Collins 1999). The order in which a parse tree is generated is crucial, since the probability of a decision to generate part of the tree can only be conditioned on structure that has already been built. Collins (1997, 1999) argues for a head-centred derivation of a tree, in which a lexical head is generated before any structure dependent on the head. He motivates this order by noting that lexical heads have a large influence on their 'locality', both in terms of the head's lexical dependents, but also the local syntactic structure.

We also use a top-down, head-centred stochastic process, but generate dependency structures rather than phrase structure trees; that is, the structures do not contain syntactic constituents as such, but simply the dependencies that exist between lexical items. The structures that are generated are similar to those produced by Link Grammar (Lafferty, Sleator, and Temperley 1992), Arc Pair Grammar (Johnson and Postal 1980), and dependency grammars in general (Melcuk 1988). The main reason for using dependency structures is that we wanted to define the probability model in terms of parameters which, as far as possible, could be estimated using the method developed in Chapter 3. In addition, by focusing on lexical sense preferences as the main source of disam-

[1] A notable exception is the work of Briscoe and Carroll (1993), who adopt a purely structural approach based on the moves of an LR parser.

biguating information, it was possible to test whether accurate parse selection could be achieved using preferences alone.

Not all the parameters of the dependency model could be estimated using the WordNet techniques. Non-nominal dependents were a problem because the WordNet estimation techniques have only been applied to a noun hierarchy, and thus an alternative estimation method was needed for these dependents. Other problems were encountered; for example, a parse selection system needs to deal with nouns, and yet the WordNet techniques have been developed to estimate the probabilities of noun senses. As each additional problem arose, we attempted to deal with it using an appropriate solution; however, the many alternative solutions were not evaluated in each case, since we did not want this research to become dominated by the parse selection problem.

The results of this chapter are a little negative, in that the dependency model fails to outperform the purely structural approach of Briscoe and Carroll. This result means that the chapter does not offer a convincing evaluation of the WordNet estimation techniques, since a possible conclusion is that accurate parse selection cannot be achieved using preferences alone, whatever estimation method is used. However, arriving at this tentative conclusion is worthwhile in itself, and a further contribution of this chapter is that we have extended similar work that uses WordNet for structural disambiguation.

The next section describes the dependency structures, and Section 5.2 describes how they are generated probabilistically, together with the independence assumptions leading to a probability model. Also, the methods for parameter estimation are described. Section 5.3 describes an implementation, and, finally, Section 5.4 gives some empirical results.

5.1 Dependency structures

The dependency structures are derived from a pre-determined set of grammatical relations, where a grammatical relation specifies the syntactic dependency that holds between a head and a dependent (Carroll, Briscoe, and Sanfilippo 1998a). More specifically, a dependency structure is a kind of labelled dependency tree, with lexical items at the nodes and grammatical relations labelling the edges. These structures lie somewhere between syntax and semantics, in that the relations may encode whether the dependent has undergone a transformational process such as passivisation or dative shift.[2] Including transformations is useful because the selectional preferences of a verb for a passive subject, for example, are likely to differ from that for an active subject. It also allows us to model the fact that some verbs are more likely to be subject to a particular transformational process than others.

Each parent and child in the tree is in a head dependent relationship, and each edge is labelled with a (r,t,f) triple, where r is a grammatical relation, t is a preposition or complementiser introducing the dependent, and f is a transformation. Following Carroll et al. (1998a), we will sometimes use the word *type* to refer to a word introducing the dependent, and use "_" in cases where there is no preposition or complementiser, or no transformation. For some relations, t or f may not be applicable, in which case the value is always "_". This notion of dependency structure is based on the representation of grammatical relations used by Carroll et al. (1998a) and Carroll et al. (1999), although the formulation in terms of a labelled graph is novel.

Figure 5.1 shows a dependency structure for an example sentence (adapted from an example in Carroll et al. 1999.) The grammatical relations are a subset of those used by Briscoe and Carroll, and at this stage are used merely as examples of what kind of relations might be applicable. Each relation is described briefly below; a more comprehensive set of examples is given in Appendix A, where the complete set of relations used in the experiments is described.

- cmod denotes a clausal modifier, and there are two examples of this relation: *become when die* (*When* the proprietor *dies*, the establishment should *become* a corporation ...); and

[2] The use of 'transformation' in this chapter is metaphorical, and does not imply a commitment to transformational theories of syntax.

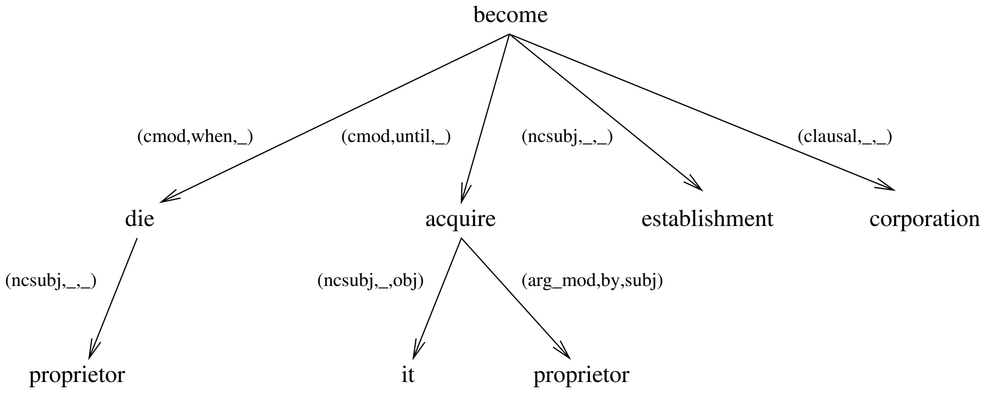

Figure 5.1: Example dependency structure for the sentence: *When the proprietor dies, the establishment should become a corporation until it is acquired by another proprietor.*

become until acquire (the establishment should *become* a corporation *until* it is *acquired* by another proprietor). Here, *become* is the head in both cases, and *die* and *acquire* are dependents. The prepositions *when* and *until* introduce the dependents.

- ncsubj denotes a non-clausal subject. The ncsubj examples simply encode a head and dependent, except that the passive *it is acquired* is recognized as such by the symbol obj. This appears in the triple labelling the edge (*acquire,it*), and indicates that *it* is an underlying object of *acquire*.

- arg_mod indicates an argument that is realised as a modifier. The example is *acquired by proprietor*, in which the modifier *proprietor* is an underlying subject of *acquire*. This is indicated by the symbol subj in the triple labelling the edge (*acquire,proprietor*). This triple also has the preposition *by* which introduces the dependent.

- clausal denotes a clausal complement.

Note that the literature does not contain a set of grammatical relations that has been agreed to be the 'correct' set, and we make no commitment to any one set here. Yeh (2000a, 2000b) notes that different sets of relations are useful for different purposes. In practice, the relations are likely to be chosen manually, based on the intuitions of a linguist. Ideally, the chosen set should lead to the probability model exhibiting what Collins (1999) calls *discriminative power* and *compactness*. Discriminative power relates to how well a set of parameters is able to lead to correct disambiguation decisions, and compactness relates to the number of parameters. There is usually a trade-off between these two requirements, in that increasing the discriminative power of a model typically increases the number of parameters. For example, using relations that encode nominal modifiers will increase the discriminative power (because noun-noun compound ambiguities can be recognized, for example), but will greatly increase the number of parameters, and hence reduce compactness.

5.2 Generating dependency structures using a history-based model

A dependency structure is generated using a history-based model, which is simply a sequence of decisions $\langle d_1, d_2, \ldots, d_n \rangle$ that generates the structure in some canonical order (Collins 1999). The probability of a dependency structure π can then be written as:

$$p(\pi) = p(\langle d_1, d_2, \ldots, d_n \rangle) = \prod_{i=1}^{n} p(d_i | d_1 \ldots d_{i-1}) \tag{5.1}$$

```
Generate the non-dependent heads, $\Theta$
for each head in $\Theta$ do
    Generate a bag of grammatical relations
    for each relation in bag do
        Generate a transformation
        Generate a dependent and type introducing the dependent
    end
end
until the leaves of the generated structure are all null dependents do
      for each non-null leaf dependent do
          Generate a bag of grammatical relations
          for each relation in bag do
              Generate a transformation
              Generate a dependent and type introducing the dependent
          end
      end
end
```

Figure 5.2: Sequence of decisions generating a dependency structure

The dependency structure with the highest probability is chosen as the correct structure (together with the corresponding parse, if necessary). The conditioning context $d_1 \ldots d_{i-1}$ is known as the history, and is equivalent to the structure built up to that point. In order that the model have a manageable number of parameters, a function Φ is used to group the histories into equivalence classes, which defines the independence assumptions in the model:

$$p(\pi) = p(\langle d_1, d_2, \ldots, d_n \rangle) = \prod_{i=1}^{n} p(d_i | \Phi(d_1 \ldots d_{i-1})) \tag{5.2}$$

A key question in defining any history-based model is how to define Φ. This question is answered below, after the decisions used to generate a dependency structure have been described. The sequence of decisions is shown in Figure 5.2, and is based on a top-down derivation of the tree.

As an example, consider how the sequence of decisions can be used to generate the dependency structure in Figure 5.1. First, the non-dependent heads are generated (we use Θ to denote the set of non-dependent heads). The term 'non-dependent head' is used to refer to a head that does not itself appear as a dependent (those heads at the 'top' of the tree). So that the tree has only one root, a single root can be generated with probability 1, followed by each non-dependent head. These are generated by first choosing the *number* of non-dependent heads, and then choosing the heads themselves. In this example, there is only one non-dependent head, so first the number 1 is chosen, and then the head *become*.

Next, a bag of relations is generated for each head. In this case, the bag {cmod, cmod, ncsubj, clausal} is generated for *become*.[3] The next stage is to generate any transformations, such as passivisation. For relations not associated with transformations, the null transformation "_" is generated with probability 1. In the example, the null transformation is generated for the ncsubj relation, indicating an active subject. Next, the dependents and, where appropriate, the types introducing the dependents are generated. Again, for relations not associated with a type, the null type "_" is generated with probability 1. At this stage, the part structure shown in Figure 5.3 has been generated.

[3] The bags of relations are similar to what Lafferty et al. (1992) call *usages* or *disjuncts*, in the context of Link Grammar.

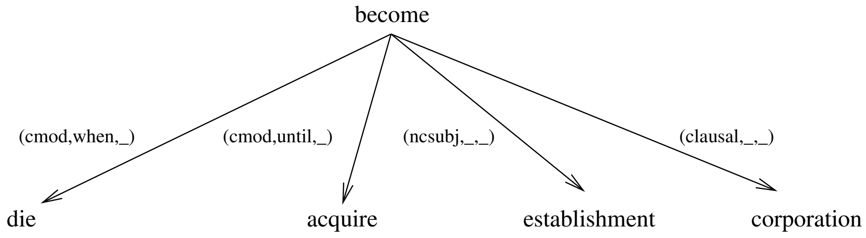

Figure 5.3: Part generation of a dependency structure

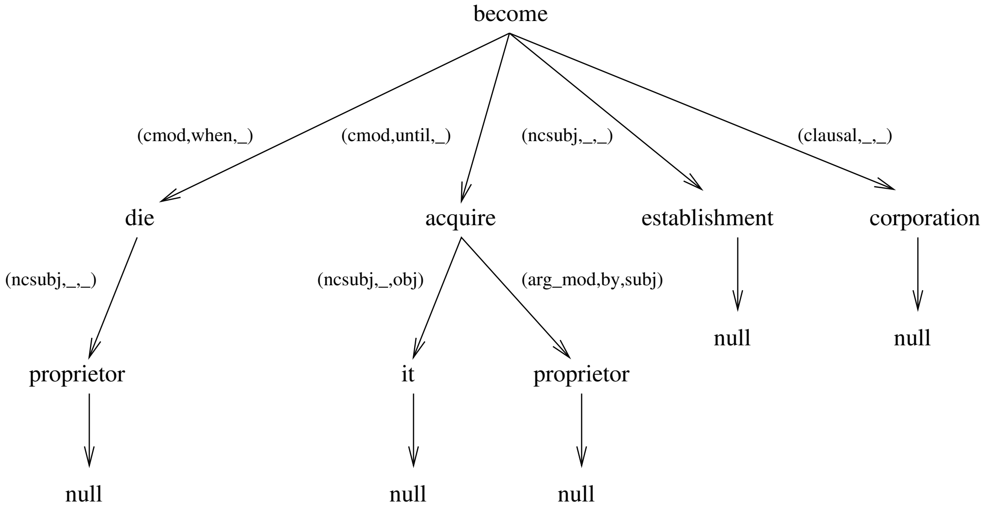

Figure 5.4: The complete dependency structure

Now we are back to choosing a bag of relations, one for each dependent. The bag may be an empty bag, indicating that the dependent is not modified in any way. The empty bags were not shown in Figure 5.1, but can be included by creating edges consisting of a non-modified dependent and a 'null' node. To be consistent, the edge can be thought of as labelled with the 'empty triple': (_,_,_). This notation is used in Figure 5.4. The process of generating bags of relations, transformations, dependents and types continues recursively until there are only null dependents left, eventually resulting in the structure in Figure 5.4.

5.2.1 The independence assumptions

The independence assumptions are determined by how much of the history is used in each conditioning context. Given the head-centred nature of the representation, and following Collins (1999), the natural choice is to condition on lexical heads, and as far as possible that approach is adopted here. The independence assumptions are given below, together with the different types of probability distribution present in the model.[4]

- $p(N)$: The probability of the number of non-dependent heads, N. This probability corresponds to the first decision in the generative process.
- $p(h)$: The probability of generating a non-dependent head h. Each head is assumed to be generated independently of the number of non-dependent heads, and independently of any previously generated head.

[4] We make a standard abuse of notation by using $p(x)$ to refer to both an individual probability and an entire probability distribution. The correct interpretation in any particular case should be clear from the context.

- $p(rb|h)$: The probability of generating a bag of relations, *rb*, for a head *h*. This probability is assumed to be dependent on *h* only.

- $p(f|h,r)$: The probability of generating a transformation, f. This probability is assumed to be dependent on the head *h* and the relation *r*.

- $p(d,t|h,r,f)$: The probability of generating a dependent *d* and type *t*. This probability is assumed to be dependent on the head *h*, the relation *r* and the transformation f.

Before giving the expression for the probability of a dependency structure, some notation and a point of clarification are needed: note that all lexical items in a dependency structure are both heads and dependents, except the non-dependent heads and the null dependents. For example, *acquire* (in Figure 5.4) is both a head, with dependents *it* and *proprietor*, and a dependent (of *become*). Now some notation: let Θ be the set of non-dependent heads in π, H be the set of all heads in π, and $\rho(h)$ denote the bag of relations associated with head h.[5] Let E be the set of labelled edges in π, and e denote some labelled edge in E; then h_e, d_e, t_e, r_e, f_e denote the head, dependent, type, relation and transformation, respectively, associated with edge e. Given this notation, and the above independence assumptions, the probability of a dependency structure π is as follows:

$$p(\pi) = p(|\Theta|) \prod_{h \in \Theta} p(h) \prod_{h \in H} p(\rho(h)|h) \prod_{e \in E} p(f_e|h_e, r_e) \prod_{e \in E} p(d_e, t_e|h_e, r_e, f_e) \tag{5.3}$$

The equation can be broken up as follows. The product $p(|\Theta|) \prod p(h)$ is the probability of generating the set of non-dependent heads, Θ. The term $\prod p(\rho(h)|h)$ is the probability of generating all the relation bags in π, where the probability for each individual bag is conditioned on the relevant head. Recall that every node in the structure is a head, except the null dependents. The term $\prod p(f_e|h_e, r_e)$ is the probability of generating all the transformations, where the probability for each individual transformation is conditioned on the head and grammatical relation associated with the relevant edge. And finally, the term $\prod p(d_e, t_e|h_e, r_e, f_e)$ is the probability of generating all the dependents, where the probability for each dependent is conditioned on the head, grammatical relation and transformation associated with the relevant edge. Recall that every node in the structure is a dependent, except the non-dependent heads.

The next section explains how the different distributions are estimated, and the data requirements for each distribution.

5.2.2 Parameter estimation

Training data

The different distributions of the model have different data requirements. For the distributions $p(|\Theta|)$ and $p(h)$, a random sample of complete dependency structures is needed, since the estimates are based on counts of non-dependent heads, and a non-dependent head is identified by the fact that it does not appear as a dependent in the rest of the structure. The data for these distributions would ideally be in the form of manually annotated dependency structures; however, there is unlikely to be much data available in this form, and so, in practice, the output of a robust parser may have to be used (assuming there is a readily available parser that can produce dependency structures as output). For the implementation described in Section 5.3, we did use the output of a parser. The details of the parser and the data used in the implementation will be given in Section 5.3.2.

The data needed for estimating $p(rb|h)$ is a random sample of (h, rb) pairs. Such a sample can be obtained from the output of a robust parser, as long as the parser can identify the necessary range of grammatical relations. The data needed for estimating $p(f|h,r)$ and $p(d,t|h,r,f)$ is a

[5] The fact that π is a parameter could be indicated by labelling each variable with π, but for ease of notation the π is omitted.

random sample of (d,t,h,r,f) tuples. These can also be obtained from a parser, again assuming that the parser is able to identify the necessary range of grammatical relations, and the necessary range of transformations. Again, for the implementation, we used the output of a parser, which will be described in Section 5.3.2.

The following sections show how the different distributions are estimated. The dependency probabilities are considered first, divided into those that can be estimated using WordNet and those that cannot, and then the remaining parameters are considered.

Estimating dependency probabilities using WordNet

The class-based estimation method of Chapter 3 can be used for those cases where the dependent is nominal, which applies to the majority of non-clausal subjects, direct objects, and indirect objects of verbs, and some modifiers. We first consider the case where there is no type introducing the dependent, which applies to relations such as non-clausal subject and direct object. Ignoring the transformation f for the moment (this will be dealt with later), the problem is to estimate the following distribution:

$p(d|h,r)$ where d is a nominal dependent

Examples from the dependency structure in Figure 5.4 include $p(\textit{proprietor}|\textit{die},\textsf{ncsubj})$ and $p(\textit{establishment}|\textit{become},\textsf{ncsubj})$.[6] These probabilities can be estimated using the class-based method, except for one complication. The class-based method applies to the probabilities of senses, whereas the dependency structures contain words. A simple solution is adopted here, which is to apply a simple word sense disambiguation technique to obtain a sense c, and then use the probability of that sense as a proxy for the probability of the dependent d. The sense of d is chosen which maximises the probability estimate:[7]

$$c = \arg \max_{c' \in \textsf{cn}(d)} p_{sc}(c'|h,r) \tag{5.4}$$

If the nominal dependent does not appear in WordNet, and is a proper name, then the name is assumed to belong to one of the sets $\overline{\langle\texttt{person}\rangle}$, $\overline{\langle\texttt{location}\rangle}$ or $\overline{\langle\texttt{organisation}\rangle}$; that is, the name is assumed to denote a person, location or organisation. The set is chosen which maximises the probability estimate. For very common words that do not appear in WordNet, such as pronouns, new nodes can be created in the WordNet hierarchy. For the implementation described in Section 5.3, new nodes were created for the common pronouns, such as *I*, *you*, *we* etc. The synset for each node contains the corresponding pronoun, and each new node has the concept $\langle\texttt{someone}\rangle$ as a parent. Finally, for the remaining dependents not in WordNet, we take the simple approach of using an average probability value, such that the probability mass is assumed to be distributed uniformly over the concepts in WordNet.

Now consider those relations for which there is a type introducing the dependent, such as the indirect object relation (iobj). The following examples use the notation (r,t,h,d), where r is a relation, t is a type, h is a head, and d is a dependent; ncmod denotes a non-clausal modifier:

- (iobj *on place tax-payer*)
- (ncmod *before receive March*)
- (ncmod *in meeting London*)

Here the problem is to estimate the following distribution:

[6] A distinction is made between lexical items with the same form but different parts of speech, so that, in the first example, it is clear that *die* is a verb. In practice, this is achieved by treating a word as a word tag pair (using only the first letter of any tag).

[7] Recall that p_{sc} is used to denote a probability estimate obtained using the class-based method of Chapter 3, which uses '**S**imilarity **C**lasses' to estimate the probability of a sense.

$p(d,t|h,r)$ where d is a nominal dependent

The probabilities corresponding to the above examples are:

- $p(\textit{tax-payer,on}|\textit{place}, \textsf{iobj})$
- $p(\textit{March,before}|\textit{receive}, \textsf{ncmod})$
- $p(\textit{London,in}|\textit{meeting}, \textsf{ncmod})$

Again, the sense of d is chosen which maximises the probability estimate, and $p(c,t|h,r)$ is used as a proxy for $p(d,t|h,r)$, where c is determined as follows:

$$c = \arg \max_{c' \in \mathsf{cn}(d)} p_{sc}(c',t|h,r) \tag{5.5}$$

The class-based approach can be used to obtain $p_{sc}(c',t|h,r)$, by first applying Bayes' theorem, and then conditioning on an appropriate set of concepts, as before. The only difference is that the conditional probability of h is now joint with t:

$$p(c',t|h,r) = p(h,t|c',r)\frac{p(c'|r)}{p(h|r)} \tag{5.6}$$

$$\approx p(h,t|\overline{c''},r)\frac{p(c'|r)}{p(h|r)} \tag{5.7}$$

The set $\overline{c''}$ is determined using the procedure described in Chapter 3. The only difference in applying the procedure is that, when comparing probabilities conditioned on sets of concepts, the probabilities $p(h,t|C_i,r)$ are compared, rather than $p(h|C_i,r)$ (where the C_i are the relevant daughter sets). Estimates of the probabilities in 5.7 are obtained using relative frequency estimates.

There is one remaining case that can be estimated using WordNet. This is adjectival or nominal modification of nominal heads. Examples include the following, using the same notation as above:

- (ncmod _ *burden disproportionate*)
- (ncmod _ *tax-payer Fulton*)
- (ncmod _ *jury county*)
- (ncmod _ *car red*)

Here, the adjectival or nominal modifier is the dependent, and the dependent is treated as the predicate. In the *red car* example, the intuition is that we are trying to model the kinds of concepts to which *red* can apply, and the generalisation in WordNet takes place for the nominal head. In this example, $\langle\texttt{car}\rangle$ might be represented by a class such as $\overline{\langle\texttt{vehicle}\rangle}$ or $\overline{\langle\texttt{transport}\rangle}$. The distribution to be estimated is as follows:

$p(d|h,r)$ where h is nominal, d is nominal or adjectival, and there is no type introducing d

The sense of the head is chosen which maximises the probability estimate, and $p(d|c,r)$ is used as a proxy for $p(d|h,r)$, where c is obtained as follows:

$$c = \arg \max_{c' \in \mathsf{cn}(h)} p_{sc}(d|c',r) \tag{5.8}$$

In this case, there is no need to apply Bayes theorem, since the probability is already conditioned on a concept, which is replaced with a suitable set of concepts:

$$p(d|c',r) \approx p(d|\overline{c''},r) \tag{5.9}$$

The set $\overline{c''}$ is obtained by applying the procedure described in Chapter 3, and the probability $p(d|\overline{c''},r)$ is estimated using relative frequencies. If the head does not appear in WordNet, an estimate of $p(d|\overline{\langle\texttt{root}\rangle},r)$ is used, unless the head is a pronoun or proper name. If the head is a pronoun, $\overline{c''}$ is set to $\overline{\langle\texttt{person}\rangle}$, and if the head is a proper name, $\overline{c''}$ is set to $\overline{\langle\texttt{person}\rangle}$, $\overline{\langle\texttt{location}\rangle}$ or $\overline{\langle\texttt{organisation}\rangle}$. In the latter case, the set is chosen which maximises the probability estimate.

To see how transformations are dealt with, consider the example of a non-clausal subject after passivisation. The subject of a passive clause and the direct object of an active clause could be treated entirely separately, but, in order to further reduce the sparse data problem, an alternative approach is taken. We assume that the conditional probability of a concept appearing as a passive subject of a verb is the same as the conditional probability of the concept appearing as a direct object:

$$p(c|v,\mathsf{ncsubj},\mathsf{passive}) = p(c|v,\mathsf{dobj},\_) \tag{5.10}$$

This approach has the advantage that the data for passive subjects and direct objects can be pooled. A similar approach can be applied to other transformations such as dative shift.

Estimating the remaining dependency probabilities

The remaining dependency probabilities correspond to relations such as clausal complementation and clausal modification. Following Carroll et al. (1998a), it is assumed that the dependent in such cases is the head of the clause, which means that the dependent is usually verbal. Carroll et al. give the following example of a sentence containing a clausal complement: *he ate the cake because he was hungry*. In this case, *eat* is the head, *be* is the dependent, and *because* is the type introducing the dependent. Some examples of clausal modifiers were given in Figure 5.1.

Since the WordNet techniques have been designed for nominal arguments only, an alternative estimation method is required for clausal complements and modifiers. The method we use is a form of linear interpolation. For those relations for which there is no type introducing the dependent, such as the clausal subject relation, the probabilities are estimated as follows ($\tilde{p}$ denotes an interpolated estimate):

$$\tilde{p}(d|h,r) \;=\; \lambda_r(h)\hat{p}(d|h,r) \;+\; (1-\lambda_r(h))\hat{p}(d|r) \tag{5.11}$$

where $0 \leq \lambda_r(h) \leq 1$. The individual estimates $\hat{p}(d|h,r)$ and $\hat{p}(d|r)$ are relative frequency estimates, calculated as follows:

$$\hat{p}(d|h,r) = \frac{f(d,h,r)}{f(h,r)} \qquad\qquad \hat{p}(d|r) = \frac{\sum_{h'\in H} f(d,h',r)}{\sum_{h'\in H} f(h',r)} \tag{5.12}$$

where $f(d,h,r)$ is the number of times dependent d appears in position r of head h, $f(h,r)$ is the number of times any dependent appears in position r of head h, and H is the set of possible heads.

The intuition behind the interpolated estimate is that if $f(h,r)$ is high, the estimate should be largely based on $\hat{p}(d|h,r)$; but if $f(h,r)$ is low, the estimate is largely based on $\hat{p}(d|r)$. This reasoning is the basis for the calculation of $\lambda_r(h)$, which is adapted from a method described in Collins 1999:

$$\lambda_r(h) = \frac{f(h,r)}{f(h,r)+\delta} \tag{5.13}$$

where δ is a positive constant that can be optimised on held-out data. This formula has the desirable characteristics that $\lambda_r(h)$ increases as $f(h,r)$ increases (approaching a maximum of 1 as $f(h,r)$ gets very large), and $\lambda_r(h)$ is 0 when $f(h,r)$ is 0.

Now consider the case where there is potentially a type introducing the dependent, which applies to clausal complementation and modification. We make the simplifying assumption that d and t are conditionally independent, so that $p(d,t|h,r)$ can be estimated as follows:

$$\tilde{p}(d,t|h,r) = \tilde{p}(d|h,r) \times \tilde{p}(t|h,r) \tag{5.14}$$

The probability $\tilde{p}(d|h,r)$ is estimated as above, and linear interpolation is also used for $\tilde{p}(t|h,r)$:

$$\tilde{p}(t|h,r) = \lambda_r(h)\,\hat{p}(t|h,r) + (1-\lambda_r(h))\,\hat{p}(t|r) \tag{5.15}$$

The individual relative frequency estimates are estimated in the same way as for $\tilde{p}(d|h,r)$.

Estimating the remaining parameters

The next probabilities to consider are $p(|\Theta|)$ and $p(h)$. The probability $p(|\Theta|)$ corresponds to the first decision in the generative process, and is the probability of generating $|\Theta|$ non-dependent heads. The probability $p(h)$ is the probability of generating a non-dependent head h. These probabilities are estimated using relative frequencies:

$$\hat{p}(N) = \frac{f(N)}{\sum_{N' \in \aleph} f(N')} \tag{5.16}$$

where $f(N)$ is the number of dependency structures with N non-dependent heads, and $\aleph$ is the set of natural numbers;

$$\hat{p}(h) = \frac{f(h)}{\sum_{h' \in H} f(h')} \tag{5.17}$$

where $f(h)$ is the number of times h appears as a non-dependent head, and H is the set of possible heads.

This leaves $p(rb|h)$ and $p(f|h,r)$. Since these probabilities are conditioned on heads, a relative frequency estimate would not be appropriate, and so linear interpolation is used:

$$\tilde{p}(rb|h) = \mu(h)\,\hat{p}(rb|h) + (1-\mu(h))\,\hat{p}(rb) \tag{5.18}$$

where $\mu(h) = f(h)/(f(h)+\varepsilon)$, and $f(h)$ is the number of times h appears as a head. Note that the constant ε is different to the δ used earlier. Since there are less types of 'relation bags' than non-nominal dependents (or at least less types of relation bags with a non-negligible chance of appearing in the data), we would expect $\varepsilon < \delta$. As before, ε can be optimised on held-out data. The individual relative frequency estimates are as follows:

$$\hat{p}(rb|h) = \frac{f(rb,h)}{f(h)} \qquad \hat{p}(rb) = \frac{\sum_{h' \in H} f(rb,h')}{\sum_{h' \in H} f(h')} \tag{5.19}$$

where $f(rb,h)$ is the number of times h appears with the bag of relations rb, $f(h)$ is the number of times h appears as a head, and H is the set of possible heads. Note that the set of possible relation bags includes the 'null bag', which is generated when a head has no dependents.

The estimate for $p(f|h,r)$ is based on the same technique:

$$\tilde{p}(f|h,r) = \mu_r(h)\,\hat{p}(f|h,r) + (1-\mu_r(h))\,\hat{p}(f|r) \tag{5.20}$$

where $\mu_r(h) = f(h,r)/(f(h,r)+\varepsilon)$. The individual relative frequency estimates are as follows:

$$\hat{p}(f|h,r) = \frac{f(h,r,f)}{f(h,r)} \qquad \hat{p}(f|r) = \frac{\sum_{h' \in H} f(h',r,f)}{\sum_{h' \in H} f(h',r)} \tag{5.21}$$

where $f(h,r,f)$ is the number of times h appears with transformation f and relation r, $f(h,r)$ is the number of times h appears with relation r, and H is the set of possible heads.

Even though the estimation techniques have been designed with low count events in mind, there may still be some zero probability estimates. As a final method to remove zero estimates, we use the fairly crude technique of "add-1/2". A more sophisticated method could have been used as a final form of smoothing, but this would not have been entirely satisfactory; as Pereira et al. (1993) comment, in the context of their clustering method, "smoothing zero frequencies appropriately ... is not very satisfactory because one of the goals of our work is precisely to avoid the problems of data sparseness by grouping words into classes."

The next section describes the parser and the set of grammatical relations used in the evaluation.

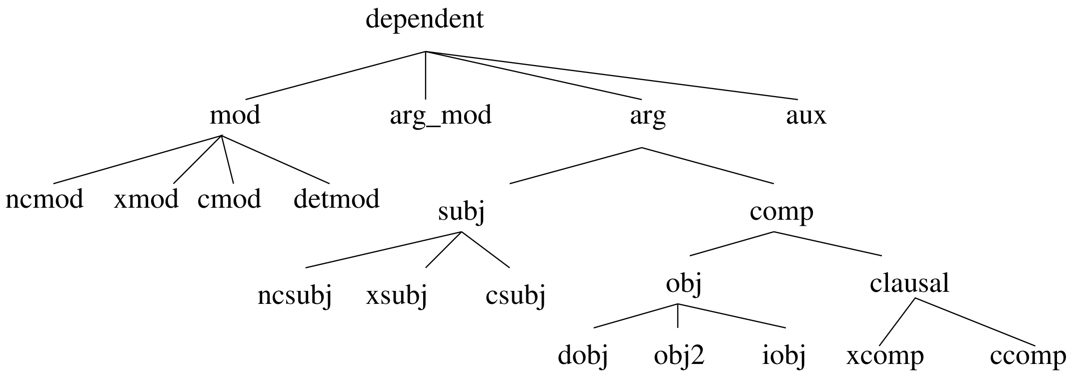

Figure 5.5: The grammatical relations used in the implementation

5.3 Implementation

5.3.1 The parser and grammatical relations

The parser used for the evaluation is a more developed version of that described in Carroll and Briscoe 1996. This version is able to produce output in the form of grammatical relations, which is the main reason the parser was chosen. The parser produces a set of parses for a sentence, together with the corresponding sets of grammatical relations. Thus we were able to create a dependency structure for each parse, and choose the parse with the most probable structure. A further advantage in using this parser is that there exists a manually created test suite which uses the same grammatical relation scheme as used by the parser (Carroll et al. 1998a, 1999); this test suite was used for the evaluation.

The relations used by the parser can be arranged in a hierarchy, as shown in Figure 5.5. If the parser is unable to determine the precise nature of the relation, and thus cannot return a relation at a leaf node, a more generic relation can be returned. Each relation is described in detail in Appendix A, based on the descriptions given in Carroll et al. 1998a and Carroll et al. 1999. A brief description of each relation is given below.[8]

- mod: relation between a head and modifier.
- ncmod, xmod, cmod: non-clausal and clausal modification; c and x indicate different control alternatives.
- detmod: relation between a noun and determiner.
- arg_mod: relation between a head and a semantic argument realised as a modifier.
- subj: relation between a predicate and its subject.
- ncsubj, xsubj, csubj: non-clausal and clausal subjects; c and x indicate different control alternatives.
- obj: relation between a head and an object.
- dobj: relation between a predicate and its direct object.
- obj2: relation between a predicate and the second non-clausal complement.

[8] Note that the version of the parser used here returns the relations detmod and aux, which are not mentioned in the Carroll et al. papers. The parser also attempts to deal with coordination, and returns the relation conj, but that relation was not used in the implementation.

```
(|ncsubj| |continue:6_VV0| |failure:1_NN1| _ )
(|clausal| _ |continue:6_VV0| |place:8_VV0|)
(|ncsubj| |place:8_VV0| |failure:1_NN1| _ )
(|dobj| |place:8_VV0| |burden:11_NN1| _ )
(|iobj| |on:12_II| |place:8_VV0| |tax-payer:14_NN2|)
(|dobj| |do:3_VD0| |this:4_DD1| _ )
(|xcomp| |to:2_TO| |failure:1_NN1| |do:3_VD0|)
(|ncmod| _ |burden:11_NN1| |disproportionate:10_JJ|)
(|ncmod| _ |tax-payer:14_NN2| |Fulton:13_NP1|)
(|detmod| _ |burden:11_NN1| |a:15_AT1|)
(|aux| _ |continue:6_VV0| |will:16_VM|)
```

Figure 5.6: Example parser output for the sentence: *Failure to do this will continue to place a disproportionate burden on Fulton tax-payer.*

- iobj: relation between a predicate and a non-clausal complement introduced by a preposition.

- clausal: relation between a head and a clausal complement.

- xcomp, ccomp: clausal complementation; c and x indicate different control alternatives.

- aux: relation between an auxiliary verb and a main or other auxiliary verb.

Figure 5.6 gives an example of the output returned by the parser, and Figure 5.7 shows the corresponding dependency structure. This is the correct structure for the sentence, although note that the structures returned by the parser may not always include the correct structure. The parser output is in the form (relation type head dependent initial_gr), where initial_gr indicates the underlying relation before any transformational process. If type or initial_gr do not apply to a particular relation, this field is omitted. The parser attempts to identify cases of passivisation and dative shift.

On the whole, the parser returns relations at the leaves of the hierarchy. However, for some relations, the parser is not always able to determine the particular control alternative, and a more generic relation is returned. A more generic relation appears in Figure 5.6, in which *continue to place* is assigned the relation clausal, rather than the more specific xcomp. This property of the parser causes a problem for the probability model. Note that if clausal were included in the model as a separate relation from xcomp and ccomp, then the model would be attempting to reflect how good the parser is at recognizing control alternatives, since clausal is only returned for those cases where the control alternative cannot be determined. This is undesirable, since the model should reflect the properties of dependency structures, and not the inadequacies of the parser. In response to this problem, the distinction between xcomp and ccomp is ignored, and all clausal complements are associated with the single relation clausal. Unfortunately, a similar problem arises with the relations mod and subj. The parser is sometimes unable to distinguish between ncmod, xmod and cmod, and also between xsubj and csubj. To deal with this problem, the three types of modification are treated as the same relation, mod, and no distinction is made between the two types of clausal subject.

The parser also fails occasionally in the identification of types introducing dependents. This occurs in the *continue to place* example, in which "_" is returned instead of *to*. (Sometimes there really is no type introducing the dependent, in which case the parser would be correct to return "_".) This problem is dealt with by assuming that the parser is always correct when returning "_".

The structure in Figure 5.7 highlights another problem that arises from using this parser with the dependency model. Dependency structures have been defined as trees, and yet the structure

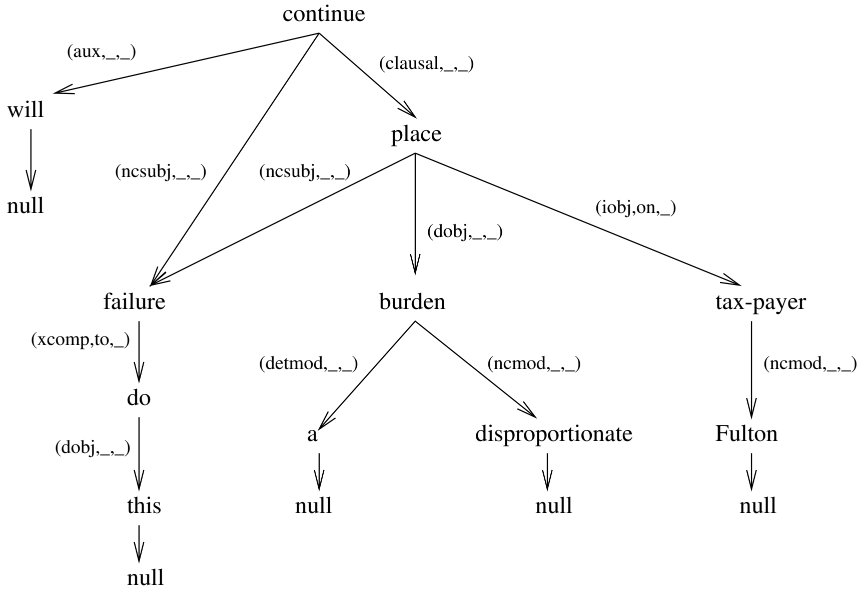

Figure 5.7: Example dependency structure for the sentence: *Failure to do this will continue to place a disproportionate burden on Fulton tax-payer.*

in Figure 5.7 is a DAG, since *failure* is a dependent of both *continue* and *place*. The fact that the parser is able to deal with control structures in this way is a positive feature of the parser, but causes a problem for the probability model. A problem arises because the generative process applied to the structure in Figure 5.7 would generate *failure* twice: once as a dependent of *continue*, and once as a dependent of *place*, but the model is unable to capture the fact that the same instance of *failure* is being generated in both cases.[9] We adopt a pragmatic solution to this problem by not modifying the model, and applying 5.3 directly to the DAG. This accounts for the fact that *failure* is a subject of both *continue* and *place*, but 5.3 does not define a proper probability distribution over the dependency structures returned by the parser.

5.3.2 Parameter estimation in practice

Obtaining training data

Section 5.2.2 explained why complete dependency structures are required for estimating some of the parameters, and how, ideally, the dependency structures would be manually annotated. There does exist a marked-up set of sentences that can be used to create dependency structures, but the set contains only 500 sentences from the Susanne corpus, and these form the test suite for the evaluation. Some of this annotated data could have been used for training, but this would have reduced the size of the test suite, and produced only a small number of structures for training. Alternatively, there is a readily available parser that can identify the necessary grammatical relations, namely the Briscoe and Carroll parser which is being used for the evaluation, and the output from the parser can be used to create dependency structures. Thus it was decided to use this parser for supplying all the training data. Note that the parser ordinarily returns just one parse for a sentence, and it is this version that is being used to supply the data; for the evaluation, *all* the parses for a sentence are considered, and the dependency model is used to select a parse.

Some of the data supplied by the parser is inaccurate, but the advantage in using automatically acquired, as opposed to manually annotated, data is that a large volume of data can be produced; the hope is that the large volume will offset the relatively low accuracy. The training data were

[9] Abney (1997) discusses a similar problem in the context of stochastic attribute-value grammars.

obtained from John Carroll, who ran the parser over around 15 million words of the BNC, from around 830,000 sentences. The parser output was in the same form as that given in Figure 5.6, and the output was processed in the following way (the formulaic expressions, such as sums of money, were found using simple regular expressions):

- 4-digit numbers beginning '1' or '2' were replaced with the word *twelvemonth*.

  Numerical expressions were replaced with *definite_quantity*.

  Monetary expressions not in WordNet were replaced with *sum_of_money*.

  Expressions denoting people not in WordNet (such as 'Dr') were replaced with *someone*.[10]

  Expressions denoting companies not in WordNet (such as 'Ltd') were replaced with *company*.

- Verbs and prepositions were reduced to lower case.

- All words were lemmatized.

The formulaic expressions were replaced with these particular words because each word has only one sense in WordNet, and belongs to a relevant synset.

Some parts of the data are much more accurate than others. Table 5.3 in the next section gives results for the Briscoe and Carroll parser, by relation. The table indicates that the data for non-clausal subjects and direct objects, for example, will be much more accurate than the data for indirect objects and second objects. The very low precision figures for iobj and obj2, and the higher recall figures, suggest that the parser is over-generating these relations. As well as providing inaccurate data for the estimation of dependency probabilities, this is a problem for the estimation of $p(rb|h)$, the probability of relation bags. By inspection, we found that many of the relation bags implied by the data erroneously contained iobj and obj2 relations, so that the estimation of $p(rb|h)$ and $p(rb)$, for a bag containing iobj or obj2, was invariably too high.

We examined 500 instances of iobj and obj2 relations from the training data, and found that 55% of the iobj cases were incorrect; more specifically, 29% should have been mod. For obj2, 80% of the cases were incorrect. To try and improve the counts for the relation bags, we reduced the count for any bag of relations containing iobj by 55%, and increased the count for the corresponding bag without an iobj, and for the bag with an iobj replaced by mod. For any bag containing obj2, we reduced the count by 80%, and increased the count for the corresponding bag without obj2. To make this clear, consider the following example. Suppose that head h occurred with the bag $\{\mathsf{ncsubj}, \mathsf{dobj}, \mathsf{iobj}\}$ 100 times in the data. The count for $(h, \{\mathsf{ncsubj}, \mathsf{dobj}, \mathsf{iobj}\})$ would be reduced to 45; the count for $(h, \{\mathsf{ncsubj}, \mathsf{dobj}, \mathsf{mod}\})$ would be increased by 29; and the count for $(h, \{\mathsf{ncsubj}, \mathsf{dobj}\})$ would be increased by 26. Now suppose head h occurred with the bag $\{\mathsf{ncsubj}, \mathsf{dobj}, \mathsf{obj2}\}$ 50 times. The count for $(h, \{\mathsf{ncsubj}, \mathsf{dobj}, \mathsf{obj2}\})$ would be reduced to 10, and the count for $(h, \{\mathsf{ncsubj}, \mathsf{dobj}\})$ would be increased by 40. The new counts were found to improve the results.

Some of these parser errors may look serious, but to put them in perspective, a fairly large percentage (77%) of the grammatical relations in the test suite are cases of ncmod, detmod, ncsubj and dobj, for which the accuracy of the data is reasonable.

Estimating the dependency probabilities

As far as possible we have tried to estimate the dependency probabilities using the class-based method of Chapter 3. This applies to nominal dependents and covers the relations ncsubj, dobj, obj2, iobj, arg_mod and some cases of mod. The extent to which this covers the relation hierarchy is shown in Figure 5.8, where the relations covered by WordNet are in the boxes. The relation mod

[10] 'Dr.' is in WordNet, but not 'Dr' (without the period). A similar comment applies to other formulaic expressions such as 'Ltd'.

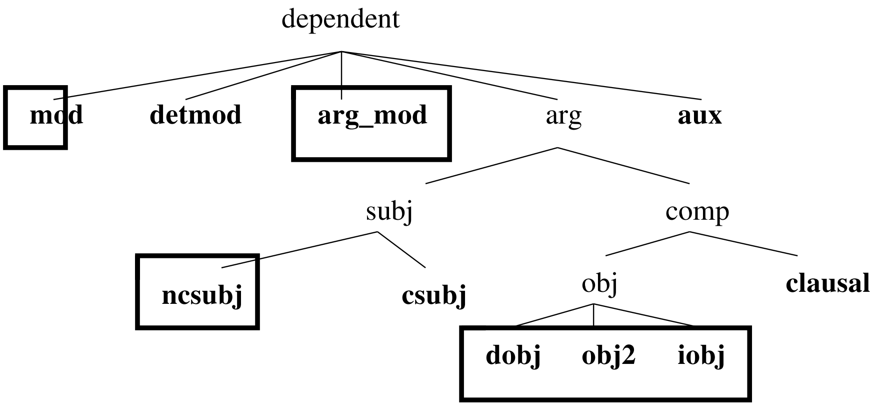

Figure 5.8: Dependency probabilities, by relation, that can be estimated using WordNet

is half covered by a box because not all of the mod cases can be estimated using WordNet. For the test suite used for the evaluation, approximately 60% of the grammatical relations correspond to parameters that can be estimated using WordNet. The parameters corresponding to the remaining relations were estimated using the linear interpolation method.

5.4 Evaluation

The test suite consists of 500 sentences taken from the Susanne corpus, covering a number of written genres and manually annotated with grammatical relation information.[11] The test suite is described in detail in Carroll et al. 1999, and much of the discussion here is based on that paper. The grammatical relation scheme used for the annotation is the same as that described in Section 5.3.1, and includes the additional relations detmod, aux and conj, which are not mentioned in the Carroll et al. paper. Note that some of the relations, such as conj and ncmod, were not used as part of the dependency model (recall that ncmod relations were treated as mod); however, these relations were still returned by the parser, and so were used as part of the evaluation.

The frequency of each relation in the test suite is shown in Table 5.1. The frequency of a relation includes the frequencies of subsumed relations, so that the number for mod, for example, includes the frequencies of ncmod, xmod, cmod and detmod. (There are only 21 cases marked as mod; the remaining 3,895 modification cases are marked up using the most specific relations.) The modifier/argument split is around 60/30 (ignoring aux and conj), and the most prevalent relation is ncmod (among those at the leaves of the relation hierarchy), followed by detmod, ncsubj and dobj. These relations account for over 3/4 of the relations in the test suite. The average number of relations per test sentence is 13.1.

In order to evaluate the dependency model, we took up to the top 1,000 parses returned by the parser for each sentence. It was not possible to take them all, due to time and space constraints, although many sentences (422) had less than 1,000 parses; the average number of parses per sentence was 280. Dependency structures were created for each parse, using the grammatical relations returned by the parser, and the dependency model was used to choose the most probable dependency structure for each sentence.

We investigated whether selecting the parse on the basis of the geometric mean of the probability, averaged according to the number of edges in the graph, improved the results. This practice is theoretically less satisfactory than using a probability model, but, in fact, the results did improve slightly (and these are the results that are presented). The improvement could have arisen because of a problem with the probability model, or because the parser was returning some 'incomplete'

[11] The test suite and accompanying evaluation software are publically available and were obtained from: http://www.cogs.susx.ac.uk/lab/nlp/carroll/carroll.html

Relation	# occurrences	% occurrences
dependent	6537	100.0
mod	3916	59.9
ncmod	2434	37.2
xmod	129	2.0
cmod	208	3.2
detmod	1124	17.2
arg_mod	41	0.6
arg	2037	31.2
subj	1047	16.0
ncsubj	1039	15.9
xsubj	5	0.1
csubj	3	0.0
comp	990	15.1
obj	586	9.0
dobj	409	6.3
obj2	19	0.3
iobj	158	2.4
clausal	404	6.2
xcomp	323	4.9
ccomp	81	1.2
aux	379	5.8
conj	164	2.5

Table 5.1: Frequency of each type of relation in the test suite

structures. The model is likely to prefer incomplete structures with a small number of relations, because in these cases less probabilities are multiplied together to get a total probability for the dependency structure.

The dependency structures were processed in similar ways to the data, in that each word was lemmatized, and formulaic expressions were replaced with words in WordNet, as described in Section 5.3.2. Because there is only a small amount of data in the test set, we did not use any of it as held-out data, and the various parameters were selected by hand. The parameters δ and ε, described in Section 5.2.2, were set to 1,000 and 50 respectively, and the level of significance for the chi-squared test, α, was set to 0.05. The results appear to be fairly robust with respect to changes in these parameters. The scores for all relations remained the same for an α value of 0.3, as opposed to 0.05, and the overall F-score varied by less than 0.2 when the values for δ and ε were increased and decreased by 50%.

For each test sentence, the grammatical relations from the most probable dependency structure were compared with the gold standard, using the evaluation software supplied with the test suite. The software computes precision (P), recall (R) and F-score (F) of the relations compared to the gold standard. The F-score is calculated as follows: $F = 2 \times P \times R/(P+R)$. Carroll et al. (1998a, 1999) motivate these dependency based measures and argue they overcome some of the shortcomings of traditional parser evaluation techniques such as 'PARSEVAL' (Harrison et al. 1991). Following Carroll et al., relations are in general compared using an equality test, except that the subject, clausal, and modifier relations are allowed to be returned, rather than the more specific ones they subsume; and a null type "_" is allowed for the modifier, clausal, and iobj relations, even if there is a type introducing the dependent.

Table 5.2 gives some lower and upper bounds for the task. The table is in the same form as that returned by the evaluation software. The scores for a relation include the subsumed relations, so that the scores for dependent, for example, relate to all the relations, and hence are the overall scores for the task. The column labelled #GRs gives the number of times a particular relation was returned as part of the chosen parse (including subsumed relations).[12] The scores for the lower

[12] The scores for arg_mod are zero because of a fault in the parser, which classified arg_mod relations as iobj. This

	Lower Bound				Upper Bound			
Relation	Precision	Recall	F-score	# GRs	Precision	Recall	F-score	# GRs
dependent	66.4	67.5	66.9	6651	82.1	82.1	82.1	6538
mod	66.4	65.4	65.9	3856	85.1	80.5	82.8	3706
ncmod	64.6	61.0	62.8	2301	85.1	79.8	82.3	2282
xmod	44.4	28.2	34.5	82	86.3	53.5	66.0	80
cmod	48.0	25.7	33.4	111	71.3	37.0	48.7	108
detmod	90.3	86.8	88.5	1081	95.6	93.6	94.8	1096
arg_mod	0.0	0.0	0.0	0	0.0	0.0	0.0	0
arg	63.6	71.1	67.2	2277	76.8	86.2	81.2	2286
subj	69.1	78.8	73.6	1194	78.9	88.4	83.4	1174
ncsubj	74.8	79.4	77.0	1103	82.5	88.7	85.5	1117
xsubj	80.0	20.0	32.0	10	100.0	80.0	88.9	4
csubj	0.0	0.0	0.0	14	0.0	0.0	0.0	12
comp	57.6	63.0	60.2	1083	74.5	83.8	79.0	1112
obj	57.8	63.5	60.5	644	74.3	87.0	80.2	686
dobj	73.7	76.3	75.0	423	83.6	88.3	85.9	432
obj2	19.4	75.8	30.8	75	42.1	84.2	56.1	38
iobj	31.5	29.2	30.3	146	61.9	84.2	71.3	215
clausal	57.3	62.2	59.7	439	75.1	79.2	77.1	426
xcomp	72.8	67.2	69.8	298	86.7	87.0	86.9	324
ccomp	47.0	42.5	44.6	73	75.0	48.2	58.7	52
aux	93.2	87.8	90.4	357	95.3	96.0	95.7	382
conj	48.8	47.9	48.3	161	57.9	57.9	57.9	164

Table 5.2: Lower and upper bounds for the F-score

bound are on the left of the table, and were obtained by randomly selecting a parse for each sentence, from those returned by the parser. The Precision, Recall and #GRs scores for each relation are averages over 10 samples, and the F-score for each relation is calculated from the average precision and recall figures. The relations with the highest F-scores are aux, detmod, ncsubj and dobj; thus it will be easier to obtain high scores for these relations, using the dependency model, compared to relations such as iobj, obj2 and the clausal complements.

The scores for the upper bound were obtained by having access to the gold standard; for each possible structure for a sentence, the overall F-score was calculated, and the structure was chosen with the highest score. If the correct structure had always been among those returned by the parser, the upper bound would have been 100. However, the upper bound is not 100, probably because of faults in the software that extracts relations from parses. Another possible reason is that the parser does not always return the correct parse among the top 1,000, which is likely unless the grammar underlying the parser has extremely wide coverage. The relations with the highest F-scores are aux, detmod, ncsubj, xcomp, dobj and ncmod; thus it will at least be possible for the dependency model to score high on these relations.

Table 5.3 gives the results for the Briscoe and Carroll parser, which uses a structural model based on the moves of an LR parser, together with a hand-written grammar, to select a parse. The highest scores are for the relations aux, detmod, ncsubj, dobj, xcomp and ncmod, and the lowest scores are for csubj, xmod, cmod, obj2 and iobj. The left of Table 5.4 gives the results for the dependency model. The relations with the highest scores correspond closely to the LR model, and the LR model outperforms the dependency model by a few points on most relations. The dependency model did outperform the structural model on the relations iobj and obj2.

We have to admit disappointment with the results, in that the overall score is less than that for the structural LR model. In an attempt to improve the results, the model was extended to include the fact that there is a preference in English for the dependent to attach close to the head. Collins (1999) uses such a 'distance measure' in his dependency models. A modified version of Collins'

fault also occurred with the version of the parser used for obtaining data.

Relation	Precision	Recall	F-score	#GRs
	(%)	(%)		
dependent	72.9	73.6	73.3	6590
mod	76.7	69.2	72.8	3531
ncmod	76.0	65.4	70.3	2091
xmod	63.3	24.0	34.8	49
cmod	61.5	26.9	37.5	91
detmod	93.0	91.2	92.1	1102
arg_mod	0.0	0.0	0.0	0
arg	65.6	81.0	72.5	2518
subj	75.0	83.2	78.9	1162
ncsubj	79.1	83.8	81.4	1101
xsubj	100.0	20.0	33.3	1
csubj	0.0	0.0	0.0	12
comp	57.5	78.7	66.4	1356
obj	55.7	79.7	65.6	838
dobj	78.3	82.2	80.2	429
obj2	27.8	79.0	41.1	54
iobj	32.7	73.4	45.2	355
clausal	60.2	77.2	67.7	518
xcomp	80.1	83.3	81.6	336
ccomp	58.1	53.1	55.5	74
aux	94.4	93.9	94.2	377
conj	57.3	57.3	57.3	164

Table 5.3: Results for the LR parser

Relation	Precision	Recall	F-score	#GRs		Precision	Recall	F-score	#GRs
dependent	70.3	71.3	70.8	6626		72.0	72.3	72.1	6570
mod	71.0	70.6	70.8	3891		73.0	71.1	72.0	3810
ncmod	70.6	67.4	69.0	2324		73.2	67.9	70.5	2258
xmod	47.3	33.3	39.1	91		48.5	37.2	42.1	99
cmod	47.2	28.9	35.8	127		50.8	29.8	37.6	122
detmod	93.0	89.6	91.3	1083		93.4	89.8	91.6	1080
arg_mod	0.0	0.0	0.0	0		0.0	0.0	0.0	0
arg	67.1	72.5	69.7	2200		68.4	74.6	71.4	2224
subj	71.6	78.8	75.0	1154		73.3	80.6	76.8	1153
ncsubj	77.1	79.2	78.2	1067		79.8	80.9	80.4	1054
xsubj	100.0	60.0	75.0	3		100.0	80.0	88.9	4
csubj	0.0	0.0	0.0	14		0.0	0.0	0.0	14
comp	62.2	65.8	64.0	1046		63.1	68.3	65.6	1071
obj	67.2	68.4	67.8	597		66.8	72.4	69.5	635
dobj	79.1	72.4	75.6	374		81.5	75.3	78.3	378
obj2	56.0	73.7	63.6	25		56.0	73.7	63.6	25
iobj	46.0	57.6	51.1	198		44.0	64.6	52.3	232
clausal	55.7	61.9	58.6	449		57.8	62.4	60.0	436
xcomp	71.9	65.0	68.3	292		73.0	65.3	69.0	289
ccomp	42.1	49.4	45.5	95		46.1	50.6	48.2	89
aux	90.2	89.5	89.8	376		90.7	90.0	90.3	376
conj	50.3	48.8	49.5	159		51.9	50.6	51.2	160

Table 5.4: Results for the dependency model on the left, and the dependency model with distance measure on the right

distance measure was used, such that the distance between a head and a dependent is defined as a triple, where the first element of the triple is the number of noun "chunks" between the head and dependent, the second element is the number of verb chunks between the head and dependent, and the third element indicates whether the head is to the left or right of the dependent. John Carroll supplied the noun chunker, which uses regular expressions to identify sequences of nouns and premodifiers in noun phrases (but is unable to identify post-modifiers such as prepositional phrases). The verb chunker, which was written especially for this work, also uses regular expressions to identify sequences of verbs (which may include adverbs and to-infinitive markers). To give an example, the sentence *She wanted to get it over fast but Ayling came into the room* contains the noun chunks *She, it, Ayling* and *the room*, and the verb chunks *wanted to get* and *came.*

In order to incorporate the distance measure into the probability model, the process generating dependency structures was modified to include an additional decision to generate a distance triple. The distance triple is generated after the dependents and types, and the probability of a distance triple is conditioned on the grammatical relation, since the probability of chunks appearing between a head and dependent clearly depends on the relation. For example, the probability of a noun chunk appearing between a verb and an indirect object is likely to be greater than the probability of a noun chunk appearing between a verb and a direct object. Let Δ_e be the distance triple between h_e and d_e; then using the same notation as in Section 5.2.1, the probability of a dependency structure π is as follows:

$$p(\pi) = p(|\Theta|) \prod_{h \in \Theta} p(h) \prod_{h \in H} p(\mathsf{p}(h)|h) \prod_{e \in E} p(f_e|h_e, r_e)\; p(d_e, t_e|h_e, r_e, f_e)\; p(\Delta_e|r_e) \tag{5.22}$$

The probability $p(\Delta_e|r_e)$ is estimated using relative frequencies:

$$\hat{p}(\Delta_e|r_e) = \frac{f(\Delta_e, r_e)}{f(r_e)} \tag{5.23}$$

where $f(\Delta_e, r_e)$ is the number of times some dependent is distance Δ_e from its head in position r_e, and $f(r_e)$ is the number of times relation r_e occurs in the data. These counts are obtained by determining the number of noun and verb chunks between each head and dependent in the data. The results for the new model are in the right half of Table 5.4 and show an improvement over the previous model. The overall result is now approaching that for the LR parser.

The LR parser uses a purely structural model without lexical parameters, and we had hoped to reinforce recent results that have shown that lexicalised models perform better than their purely structural counterparts (Charniak 1997). We did not achieve this, but the dependency model did perform better at identifying some relations, such as obj2 and iobj.

There are a number of possible reasons for the poor results. One obvious problem is that of obtaining quality data. Table 5.3 indicates that, for some relations, the accuracy of the training data returned by the LR parser is very low. Using treebank data is not a viable alternative because treebanks are not marked up with enough information to easily identify all the relations, and results from the next Chapter show that, for the WordNet estimation techniques to work well, more data are required than currently available in treebanks.

The relations whose parameters are estimated using linear interpolation are not well identified by the dependency model, and the scores corresponding to such relations may be improved by using a more sophisticated estimation method. One way to further this work would be to investigate if the verb taxonomy of WordNet could be used in the same way as the noun taxonomy. There is a relation similar to hyponymy in the verb taxonomy, which has been called *troponymy* (Fellbaum 1998a). Fellbaum defines verb V_1 as a troponym of verb V_2 if V_1 *is to* V_2 *in some particular manner.* Fellbaum gives the example that *march* is a troponym of *walk*, since to march is to walk in some manner. Whether the verb hierarchy could be used in this way is debatable, since it is much flatter than the noun hierarchy, and the number of levels rarely exceeds four (Fellbaum 1998a). In addition, the introduction of verb senses would add another level of sense ambiguity.

The treatment of word sense ambiguity is another area that could be improved. Currently, a rather cavalier approach is taken, which is to select the sense that maximises the relevant probability estimate. One promising approach is to try and integrate the word sense disambiguation into the parsing model, and perform the two simultaneously, as Bikel (2000) has attempted to do.

A tentative conclusion of this chapter is that the use of lexical sense preferences, or selectional preferences, alone is unlikely to lead to a highly accurate parse selection system. Even the successful statistical parsing models, such as those of Collins (1997) and Charniak (2000), which rely heavily on lexical information, also make use of the structural properties of a parse. One way to extend this work would be to try and combine the dependency model with the structural model of Briscoe and Carroll.

As an evaluation of the class-based estimation technique, the results are inconclusive, since the parse selection problem may not be a good way to isolate the performance of the WordNet estimation techniques. In order to have a more focused evaluation, the method of estimation is applied to two disambiguation tasks that can be tackled using only parameters relating to lexical sense preferences; moreover, the parameters can be estimated using reliable data. These tasks are presented in the next chapter.

Chapter 6

Ambiguity Resolution: a comparison of class-based estimation techniques

In this chapter, two task-based evaluations of the main estimation method are presented.[1] The first task is the resolution of PP-attachment ambiguities, and this is used to compare the generalisation procedure described in Chapter 3 with a very simple procedure that selects a fixed level of generalisation. The fixed level is the set of roots of the nine complete sub-hierarchies (see Figure 3.1). The second task is the pseudo disambiguation task described in Chapter 4, and this is used to compare the estimation method with alternative class-based methods using WordNet. The approaches chosen for comparison are those of Resnik (1998) and Li and Abe (1998).

6.1 Resolving PP-attachment ambiguities

6.1.1 The problem

The problem of resolving PP-attachment ambiguities has been addressed by many researchers, and Chapter 2 described a number of previous approaches. The PP-attachment problem that is usually considered takes a four-tuple, (v, n_1, pr, n_2), and the problem is to decide whether the prepositional phrase (consisting of pr and the noun phrase headed by n_2) attaches to v or n_1. Popular examples of PP ambiguity include the following:

(6.1) I saw a man with a telescope.

(6.2) I hit a man with a stick.

The problem in these cases is to decide whether *with a telescope* attaches to *saw* or *man*, and whether *with a stick* attaches to *hit* or *man*.

The stick and telescope examples are useful in providing an intuition about the problem, but, as well as being slightly contrived, can be misleading. The reason is that the ambiguity in these examples is easy to perceive. Consider the following examples, adapted from cases in the Penn Treebank:

(6.3) I left the chairmanship of the company.

(6.4) This takes us into a new world.

(6.5) It shocked analysts despite the speculation.

[1] The first evaluation has been published as part of Clark and Weir 2000, and the second has been published as part of Clark and Weir 2001.

For these examples, it is hard to see that there is an ambiguity at all, but the attachment problem assumes that any *verb np prep np* sequence results in an ambiguity. In 6.3, it is assumed that *of the company* could attach to *left*; in 6.4, *into a new world* could attach to *us*; and in 6.5, *despite the speculation* could attach to *analysts*.

Another reason why the telescope and stick examples are misleading is that they imply the PP-attachment problem, as we have defined it, is harder than it really is. For these two examples, either attachment results in a plausible semantic reading, and the correct reading depends on the wider context. In a commonly cited paper, Altmann and Steedman (1988) argue that the resolution of attachment ambiguities requires a model where the relevant entities are represented and reasoned about. This argument led Hindle and Rooth (1993) to comment that, if this is typical of PP-attachment ambiguities, then there is little hope of building computational models to solve the problem, at least in the near future.

Clearly, some account of context is required for the resolution of some cases of attachment ambiguity. However, this may only apply to a small subset of cases. The three treebank examples can be resolved without resorting to the wider context; in fact, they can be resolved without even considering n_2. In 6.3, *left* is very unlikely to be modified by a PP headed by *of*; in 6.4, *us* is unlikely to be modified by *into*; and in 6.5, *analysts* is unlikely to be modified by *despite*. In short, *left of*, *us into* and *analysts despite* are unlikely head-preposition combinations. In contrast, *chairmanship of*, *takes into* and *shocked despite* are all perfectly acceptable. This suggests a possible solution: compare the *v-pr* and n_1*-pr* combinations, and choose the one that is the most likely.

6.1.2 A probabilistic solution

A probabilistic solution along these lines would be to compare the probabilities $p(pr|n_1)$ and $p(pr|v)$. We would expect $p(\mathit{of}|\mathit{chairmanship})$ to be greater than $p(\mathit{of}|\mathit{left})$, for example. This suggestion is very similar to the original corpus-based approach of Hindle and Rooth (1993). The problem with the suggestion (as discussed in Chapter 2) is that, in some cases, n_2 provides information that is needed to resolve the ambiguity. The obvious extension incorporating n_2 is to compare the probabilities $p(pr,n_2|n_1)$ and $p(pr,n_2|v)$, and this is the approach taken here, but with a noun sense replacing n_2. Using a sense allows the WordNet estimation method to be applied. Presumably, Hindle and Rooth ignored n_2 because of the sparse data problems introduced by considering it, but our class-based method is designed to deal with such problems.

We decide on the attachment site using the following procedure:[2]

```
if $n_2$ is not in WordNet
   if $\hat{p}(pr|v) > \hat{p}(pr|n_1)$
      then attach to verb
   else attach to noun
else if $p_{sc}(c_v,pr|v) > p_{sc}(c_{n_1},pr|n_1)$
        then attach to verb
else attach to noun
```

The concepts c_v and c_{n_1} are determined as follows:

$$c_v = \arg\max_{c \in \mathrm{cn}(n_2)} p_{sc}(c,pr|v) \tag{6.6}$$

$$c_{n_1} = \arg\max_{c \in \mathrm{cn}(n_2)} p_{sc}(c,pr|n_1) \tag{6.7}$$

That is, the sense of n_2 is chosen which maximises the relevant probability estimate.

The estimates $\hat{p}(pr|v)$ and $\hat{p}(pr|n_1)$ are obtained using relative frequencies, unless $f(v)$ or $f(n_1)$ are zero, in which case the corresponding estimate is undefined, and $\hat{p}(pr)$ is used instead.

[2] Recall that p_{sc} denotes a probability estimate obtained using the class-based method of Chapter 3.

The estimates $p_{sc}(c_v,pr|v)$ and $p_{sc}(c_{n_1},pr|n_1)$ are obtained using the method described in Chapter 3. First, Bayes' rule is applied, and then probabilities are conditioned on a set of concepts where appropriate. The formulae are given for $p(c_v,pr|v)$ only, but analogous formulae hold for $p(c_{n_1},pr|n_1)$:

$$
\begin{aligned}
p(c_v,pr|v) &= p(v|c_v,pr)\frac{p(c_v,pr)}{p(v)} && (6.8)\\
&= p(v|c_v,pr)\frac{p(pr|c_v)p(c_v)}{p(v)} && (6.9)\\
&\approx p(v|[c_v],pr)\frac{p(pr|[c_v]')p(c_v)}{p(v)} && (6.10)
\end{aligned}
$$

where $[c_v] = \overline{\mathtt{top}(c_v,v,pr)}$ and $[c_v]' = \overline{\mathtt{top}(c_v,pr)}$.

Each of the probabilities in 6.10 can be estimated using relative frequencies. General formulae for these estimates were given in Chapter 3, and so are not repeated here. If $f(v) = 0$, in which case $\hat{p}(v) = 0$ and the estimate corresponding to 6.10 is undefined, an estimate of the backed-off probability $p(c_v,pr) \approx p(pr|[c_v]')p(c_v)$ is used instead.

Note that the expression in 6.10 is slightly different to the corresponding expression given in Chapter 3. If pr is treated as the argument slot r, then the probability being estimated is of the form $p(c,r|v)$, whereas in Chapter 3 the estimation method was applied to $p(c|v,r)$. This results in the previously unseen term $p(pr|[c_v]')$, where $[c_v]' = \overline{\mathtt{top}(c_v,pr)}$. However, $\mathtt{top}(c_v,pr)$ can be determined in an entirely analogous fashion to $\mathtt{top}(c,v,r)$. Recall that the generalisation procedure determines $\mathtt{top}(c,v,r)$ by comparing probabilities conditioned on daughter sets, $p(v|C_i,r)$, where C_i are the daughter sets. The only difference, when determining $\mathtt{top}(c_v,pr)$, is that the probabilities $p(pr|C_i)$ are compared instead. The procedure progresses up the hierarchy until the probabilities $p(pr|C_i)$ are significantly different.

The test and training data described in Ratnaparkhi et al. 1994 were used for the experiments. These are from the WSJ section of the Penn Treebank, and have now become the standard data for this task. The data consist of four-tuples, (v,n_1,pr,n_2), together with the attachment site for each tuple (v or n_1). There are $3,097$ test cases, and $20,801$ cases in the training data. For each case in the training data, the triple corresponding to the correct attachment site, (v,pr,n_2) or (n_1,pr,n_2), was extracted. All cases in the training data (but not test data), for which n_2 was not in WordNet, were ignored.

In order to increase the number of training triples, we took the WSJ section of the Penn Treebank, and extracted triples from unambiguous cases of attachment (where n_2 was in WordNet). This resulted in a total of $66,881$ triples. Examples of unambiguous cases of attachment include the following (adapted from sentences in the Penn Treebank):

(6.11) This *form of asbestos* has caused many deaths. (Noun attachment)

(6.12) The asbestos was *used in modest amounts*. (Verb attachment)

Both the training and test data were pre-processed using the steps given in Section 5.3.2. Note that n_1 was processed as well as n_2.

6.1.3 Results

Before presenting the results, some lower and upper bounds are given for the task. A useful lower bound is 72%, which is the score obtained when making the attachment decision on the basis of the preposition alone, by choosing the attachment that is most often associated with the given preposition in the training data. A useful upper bound is 88%, which is the average score achieved by three human judges, when shown only the four head words (although on a different, smaller

Generalisation technique	% correct
Similarity-class	80.3
Select root of sub-hierarchy	77.9
Always select ⟨`root`⟩	79.0

Table 6.1: PP results for the complete test set of 3,097 test cases

Generalisation technique	% correct
Similarity-class	90.3
Select root of sub-hierarchy	81.4
Always select ⟨`root`⟩	79.6

Table 6.2: PP results when ⟨`root`⟩ is selected for neither attachment point – 113 test cases

test set). This result was reported by Ratnaparkhi et al. (1994). The performance of the human judges went up to 93% when given the complete sentence.

The first set of results is given in Table 6.1. The G^2 statistic was used in the chi-squared test, with a significance level, α, of 0.05. The generalisation procedure, which is referred to as the 'similarity-class technique', was compared with the simple technique of using a fixed level. Two fixed levels were used: the root of the entire hierarchy, ⟨`root`⟩, and the set consisting of the roots of each of the nine sub-hierarchies. Note that always choosing ⟨`root`⟩ results in an approach very similar to that of Hindle and Rooth, since n_2 is, in effect, being ignored. We can see this by substituting $\overline{\langle\texttt{root}\rangle}$ into 6.10:

$$p(v|\overline{\langle\texttt{root}\rangle},pr)\frac{p(pr|\overline{\langle\texttt{root}\rangle})p(c_v)}{p(v)} = p(v|pr)\frac{p(pr)p(c_v)}{p(v)} \tag{6.13}$$

$$= p(pr|v)p(c_v) \tag{6.14}$$

In this case, c_v is the sense of n_2 that results in the highest probability estimate of $p(pr|v)p(c)$, where c ranges over the senses. Similarly, c_{n_1} is the sense of n_2 that results in the highest probability estimate of $p(pr|n_1)p(c)$. However, since $p(pr|v)$ and $p(pr|n_1)$ are constants over senses of n_2, $c_v = c_{n_1}$. Therefore, comparing $p(pr|v)p(c_v)$ and $p(pr|n_1)p(c_{n_1})$ is equivalent to comparing $p(pr|v)$ and $p(pr|n_1)$.

The results for similarity-class are below the state of the art, and only slightly higher than the results for ⟨`root`⟩. However, the results are comparable with those of Li and Abe (1998), who adopt a similar approach using WordNet, but with a different training and test set. They improved on a Hindle and Rooth type approach by 1.5%, which is in line with our results.

As an evaluation of the estimation method and generalisation procedure, the result is inconclusive. This is because the sparse data problems associated with PP-attachment are particularly problematic, since WordNet is being populated for *combinations* of predicates and prepositions. For many predicate-preposition combinations that occur infrequently in the data, there are few examples of n_2 that can be used for populating WordNet. To obtain some idea of the severity of this problem, we took all those test cases for which a level other than ⟨`root`⟩ was selected for both $\texttt{top}(c_v,v,pr)$ and $\texttt{top}(c_{n_1},n_1,pr)$. This only applied to 113 cases. However, for these cases, the results are very good, as shown in Table 6.2. Here, there is some information about n_2, and the similarity-class technique is able to make a difference.

Despite these good results, it is not possible to draw any strong conclusions from the performance on only 113 test cases. To increase the number of test cases, we took those cases for which ⟨`root`⟩ was selected for at most one of $\texttt{top}(c_v,v,pr)$ and $\texttt{top}(c_{n_1},n_1,pr)$. This applied to 1,032 cases. The results are shown in Table 6.3. The result for similarity-class is state of the art on a third of the test cases, and the technique of selecting a fixed level is being outperformed.

Generalisation technique	% correct
Similarity-class	88.1
Select root of sub-hierarchy	85.5
Always select ⟨root⟩	85.5

Table 6.3: PP results when ⟨root⟩ is selected for at most one of the attachment points – 1,032 test cases

α value	% correct – G^2	% correct – X^2
0.0005	79.6	79.9
0.05	80.3	80.2
0.3	80.2	80.0

Table 6.4: PP results for G^2 and X^2 on complete test set of $3,097$ cases

Two further attempts were made to improve the results. First, we applied the extraction heuristic of Ratnaparkhi (1998) to WSJ text, in order to obtain more training data. Simple regular expressions were used to locate possible cases of unambiguous attachment. This increased the number of triples by around a factor of 10, but had little effect on the results. Presumably, this is because of the noise in the resulting data; Ratnaparkhi reports only 69% accuracy for the heuristic when applied to the Penn Treebank (excluding cases where the preposition is *of*).

Second, we used counts that had been estimated using the re-estimation algorithm described in Chapter 4, but this also had no impact on the results. The reason is that the re-estimation algorithm relies on the accumulation of counts in relevant areas of WordNet. The expectation is that counts will accumulate in sets that are representative of the correct sense of a noun, but not in sets that are representative of the incorrect senses. However, if there are few data for the predicate and argument slot in question, this accumulation will not have chance to occur. A consequence of this is that the generalisation procedure, when finding representative sets, is likely to choose sets dominated by concepts high in the hierarchy. In fact, as the number of cases for Table 6.2 shows, the procedure may choose ⟨root⟩ for many predicate-preposition combinations. If this occurs, there is no way to distinguish between the alternative senses, and the count for a noun is simply divided equally among its senses.

As a final experiment, we investigated how the performance changed when using different values for α, and how the performance changed when using the X^2, rather than G^2, statistic. The results are given in Tables 6.4 and 6.5. Table 6.4 gives the results for the complete test set, and Table 6.5 the results for those cases where ⟨root⟩ was selected for at most one of $\mathtt{top}(c_v, v, pr)$ and $\mathtt{top}(c_{n_1}, n_1, pr)$. Note that the number of cases satisfying the latter condition varies for the two statistics, as shown in the table.

The results show that, for this task, the value of α makes little difference to performance. The scores are higher in Table 6.5 for lower values of α, but this is for a smaller, and potentially easier, test set in each case. The choice of statistic also makes little difference. In Clark and Weir 2000, the comment is made that G^2 was found to perform slightly better than X^2. The results for G^2 in Table 6.5 are slightly higher, although the differences may not be statistically significant, and the G^2 results are for a slightly smaller test set in each case. Therefore, the correct conclusion is that no difference in performance has been found. In the next section, the choice of statistic is investigated further using the pseudo disambiguation task.

In conclusion, we have shown that when instances of WordNet are well populated with examples of n_2, the disambiguation method is highly accurate. When WordNet is sparsely populated, the method naturally resorts to comparing just the preposition and each of the attachment sites (so n_2 is ignored). In addition, the generalisation procedure described in Chapter 3 has been shown to

α value	% correct – G^2		% correct – X^2	
0.0005	90.1	(764 cases)	89.9	(870 cases)
0.05	88.1	(1,032 cases)	87.6	(1,248 cases)
0.3	87.1	(1,447 cases)	86.8	(1,576 cases)

Table 6.5: PP results for G^2 and X^2 when $\langle\texttt{root}\rangle$ is selected for at most one attachment point

be superior to using a fixed level of generalisation.

6.2 A pseudo disambiguation task

The pseudo disambiguation task has already been described in Chapter 4, but the description is repeated here. The task is to decide which of two verbs, v and v', is more likely to take a given noun, n, as an object. The test and training data were obtained as follows. A number of verb direct object pairs were extracted from a subset of the BNC, using the system of Briscoe and Carroll (1997). All those pairs containing a noun not in WordNet were removed, and each verb and argument was lemmatised. This resulted in a data set of around 1.3 million (v,n) pairs.

To form a test set, 3,000 of these pairs were randomly selected, such that each selected pair contained a fairly frequent verb. (Following Pereira et al. (1993), only those verbs that occurred between 500 and 5,000 times in the data were considered.) Each instance of a selected pair was then deleted from the data. This was to ensure that the test data were unseen. The remaining pairs formed the training data. To complete the test set, a further fairly frequent verb, v', was randomly chosen for each (v,n) pair. The random choice was made according to the verb's frequency in the original data set, subject to the condition that the pair (v',n) did not occur in the training data.

Given the set of (v,n,v') triples, the task is to decide whether (v,n) or (v',n) is the correct pair. Note that the sampling procedure does not guarantee that the correct pair, (v,n), is more plausible than the corresponding incorrect pair, (v',n), since a highly plausible incorrect pair could be generated by chance. The assumption is that this will occur infrequently in practice.

The next section describes how we make the disambiguation decision, and how the decision is made using the alternative class-based methods of Resnik (1993a, 1998), which was subsequently developed by Ribas (1995b), and Li and Abe (1998), which has been adopted by McCarthy (2000). These have been chosen for comparison because they directly address the question of how to find a suitable level of generalisation in WordNet. Both approaches have been described in detail in Chapter 2.

6.2.1 The alternative approaches to disambiguation

Using our technique, the disambiguation decision for each (v,n,v') triple was made using the procedure in Figure 6.1. If n has more than one sense, the sense is chosen which maximises the relevant probability estimate; this explains the maximisation over $\mathsf{cn}(n)$. The probability estimates were obtained using the technique described in Chapter 3, and the G^2 statistic was used for the chi-squared test.

The first alternative to our approach is to make the disambiguation decision on the basis of the 'association score', which is a measure of how well a set of concepts, C, satisfies the selectional preferences of a verb, v, for argument position, r:[3]

$$\mathrm{A}(C,v,r) = p(C|v,r)\log_2\frac{p(C|v,r)}{p(C|r)} \tag{6.15}$$

[3] In Chapter 4 the symbol A was used to refer to the association *norm*. In this chapter, it is used to refer to the association score.

```
if $\max_{c \in \mathsf{cn}(n)} p_{sc}(c|v,\mathrm{obj}) > \max_{c \in \mathsf{cn}(n)} p_{sc}(c|v',\mathrm{obj})$
  then choose $(v,n)$
else if $\max_{c \in \mathsf{cn}(n)} p_{sc}(c|v',\mathrm{obj}) > \max_{c \in \mathsf{cn}(n)} p_{sc}(c|v,\mathrm{obj})$
  then choose $(v',n)$
else choose at random
```

Figure 6.1: Procedure for determining the correct verb in pseudo-disambiguation task

In fact, this is not quite how Resnik defines his ‘selectional association’ measure, because of the way he estimates class probabilities. Selectional association also has a term that measures how strongly the verb selects for its arguments overall, as was described in Chapter 2. The definition used here is that given by Ribas (1995b).

An estimate of the association score, $\hat{\mathrm{A}}(C,v,r)$, can be obtained using relative frequency estimates of the probabilities. The key question is how to find a suitable set for a concept, c, assuming the choice is from those sets dominated by a hypernym of c. Resnik's suggestion is to choose the set that maximises the association score. Adopting Resnik's approach, the decision for each test triple was made as follows:

```
if $\max_{c \in \mathsf{cn}(n)} \max_{c' \in \mathsf{h}(c)} \hat{\mathrm{A}}(\overline{c'},v,\mathrm{obj}) > \max_{c \in \mathsf{cn}(n)} \max_{c' \in \mathsf{h}(c)} \hat{\mathrm{A}}(\overline{c'},v',\mathrm{obj})$
  then choose $(v,n)$
else if $\max_{c \in \mathsf{cn}(n)} \max_{c' \in \mathsf{h}(c)} \hat{\mathrm{A}}(\overline{c'},v',\mathrm{obj}) > \max_{c \in \mathsf{cn}(n)} \max_{c' \in \mathsf{h}(c)} \hat{\mathrm{A}}(\overline{c'},v,\mathrm{obj})$
  then choose $(v',n)$
else choose at random
```

where $\mathsf{h}(c)$ is the set consisting of the hypernyms of c. An additional complication arises if n has more than one sense; this explains the double maximisation. The inner maximisation is over the hypernyms of c, $\mathsf{h}(c)$, assuming c is the chosen sense of n, which corresponds to Resnik's method of choosing a set to represent c. The outer maximisation is over the senses of n, $\mathsf{cn}(n)$; this determines the sense of n by choosing the sense that maximises the association score.

The MDL approach makes the disambiguation decision by comparing the probabilities $p(n|v,\mathrm{obj})$ and $p(n|v',\mathrm{obj})$. The probabilities are estimated by choosing a class to represent n, and dividing the probability of the class evenly among the nouns in the class. We described in Chapter 2 how Li and Abe use MDL to determine a ‘cut’ across WordNet, which can be used to select a class for a noun. We also showed how the structure of WordNet provides various problems for the MDL approach. One problem is that all nouns are required to be represented at leaves of the hierarchy. McCarthy's solution is adopted here, which is to form new leaf nodes for each synset appearing at an internal node. That way, every noun in WordNet appears at a leaf node.

Another problem is that Li and Abe's use of MDL only strictly applies to a tree, and WordNet is a DAG. Again, McCarthy's (2000) solution is adopted, which is to argue that, since WordNet is a close approximation to a tree, it is better to maintain the structure of WordNet and apply MDL to the DAG. However, this did create a problem, in that that many of the cuts returned by MDL were over-generalising at the ⟨entity⟩ node. That is, many cuts contained the ⟨entity⟩ node when the data suggested the cut should have been lower. The reason is that ⟨person⟩, which is close to ⟨entity⟩ and a hyponym of ⟨entity⟩, has two parents: ⟨life_form⟩ and ⟨causal_agent⟩. This DAG-like property was responsible for the over-generalisation, and so we removed the link between ⟨person⟩ and ⟨causal_agent⟩. This appeared to solve the problem, and the results presented later for the average degree of generalisation are consistent with those given by Li and Abe (1998). This suggests that our implementation is not over-generalising relative to Li and Abe's.

A final problem is that nouns can appear in more than one synset. This is dealt with by treating each occurrence of a polysemous noun in WordNet as a separate noun, in effect treating

Generalisation technique	% correct	av.gen.	sd.gen
Similarity-class			
$\alpha = 0.0005$	73.8	3.3	2.0
$\alpha = 0.05$	73.4	2.8	1.9
$\alpha = 0.3$	73.0	2.4	1.8
$\alpha = 0.75$	73.9	1.9	1.6
$\alpha = 0.995$	73.8	1.2	1.2
Low-class	73.6	0.9	1.0
MDL	68.3	4.1	1.9
Assoc	63.9	4.2	2.1

Table 6.6: Results for the pseudo disambiguation task

it as a noun, noun sense pair. For example, the two instances of *coke* in the synsets $\{\,coke\,\}$ and $\{\,cocaine, cocain, coke, snow, C\}$ are treated as separate nouns. We use $\mathsf{sep}(n)$ to denote the set of separate instances of n in WordNet.

Adopting the MDL approach, the disambiguation decision was made as follows ($\tilde{p}$ is used to denote an estimate using the MDL approach):

```
if  $\max_{n' \in \mathsf{sep}(n)} \tilde{p}(n'|v,\mathrm{obj}) > \max_{n' \in \mathsf{sep}(n)} \tilde{p}(n'|v',\mathrm{obj})$
  then  choose $(v,n)$
else if  $\max_{n' \in \mathsf{sep}(n)} \tilde{p}(n'|v',\mathrm{obj}) > \max_{n' \in \mathsf{sep}(n)} \tilde{p}(n'|v,\mathrm{obj})$
  then  choose $(v',n)$
else  choose at random
```

The instance of n is chosen which maximises the relevant probability estimate. McCarthy's (2000) implementation of MDL was used to estimate the parameters (subject to the above modification).

6.2.2 Results

The first set of results is given in Table 6.6. As before, our technique is referred to as the 'similarity-class' technique, and results are given for a range of α values. The results demonstrate clearly that the performance of similarity-class varies little with changes in α, and similarity-class outperforms both alternatives, MDL and Assoc.[4]

We also give a score for our approach using a simple generalisation procedure, which we call "Low-class". The procedure is to select the first class that has a count greater than zero (relative to the verb and argument position), which is likely to return a low level of generalisation, on the whole. The results show that our generalisation technique only narrowly outperforms the simple alternative. Note that, although "Low-class" is based on a very simple generalisation method, the estimation method is still using our class-based technique, by applying Bayes' theorem and conditioning on a class, as described in Section 5.2.2; the difference is in how the class is chosen.

In order to investigate the differences in performance, we calculated the average number of generalised levels for each approach, as described in Section 3.5. For each test case, the number of generalised levels for both verbs, v and v', was calculated, but only for the chosen sense of n. The results are shown in the third column of Table 6.6, and demonstrate clearly that both MDL and Assoc are generalising to a greater extent than similarity-class. (The fourth column gives a standard deviation figure.) Note that this result for MDL is consistent with Li and Abe 1998, in

[4] The results given for similarity-class are different to those given in Clark and Weir (2001) because the probability estimates used in Clark and Weir (2001) were not normalised.

Generalisation technique	% correct	av.gen.	sd.gen
Similarity-class			
$\alpha = 0.0005$	66.7	4.5	1.9
$\alpha = 0.05$	68.4	4.1	1.9
$\alpha = 0.3$	70.2	3.7	1.9
$\alpha = 0.75$	72.3	3.0	1.9
$\alpha = 0.995$	72.4	1.9	1.6
Low-class	71.9	1.1	1.1
MDL	62.9	4.7	1.9
Assoc	62.6	4.1	2.0

Table 6.7: Results for the pseudo disambiguation task with 1/5th training data

which a value of around 5 is given for the average number of generalised levels (although on a smaller data set, which would explain the lower figure here).

These results suggest that MDL and Assoc are over-generalising, at least for the purposes of this task. This is in contrast with the conclusions of Li and Abe, who consider the significant amount of generalisation performed by MDL to be a positive feature. In further contrast with our results, they argue that Assoc tends to overfit the data:

> One can see that a significant amount of generalization is performed by our method– the resulting tree cut is about 5 levels higher than the starting cut, on the average.
>
> Our experiments show ... that the generalization method currently employed by Resnik has a tendency to overfit the data. ... Note that MDL tends to select a tree cut closer to the root of the thesaurus tree. (Li and Abe 1998)

For the task that Li and Abe consider, the PP-attachment task, a higher level of generalisation may be more appropriate than for the pseudo disambiguation task. The experiments performed earlier on PP-attachment offered no firm conclusions either way. But for the pseudo disambiguation task, it appears that the amount of generalisation performed by MDL is adversely affecting performance, and that Assoc is also over-generalising, rather than over-fitting the data.

The advantage that our approach has over both MDL and Assoc is that the parameter, α, allows some control over the extent of generalisation. This means the approach can potentially perform well on a range of tasks, whereas MDL and Assoc can only perform well on tasks where a significant amount of generalisation is required. It is possible that the MDL encoding scheme could be modified to include a parameter with a similar effect to α, but that has not been investigated here. Wagner (2000) investigates modifying the encoding scheme, but for the purposes of acquiring selectional preferences. There is another feature of MDL that is potentially harming performance, which is the assumption that the probability mass for a class is uniformly distributed among its members, which Li and Abe recognize as a weakness. Our approach makes no such assumption.

To investigate why the value for α had no impact on the results, we repeated the experiment, but with 1/5th of the data. A new data set was created by taking every 5th pair of the original 1.3 million pairs. A test set of 3,000 triples was created from this new data set, as before, but this time only verbs that occurred between 100 and 1,000 times were considered. The results using these test and training data are given in Table 6.7.

These results show a variation in performance across values for α, with an optimal performance when α is around 0.75. (Of course, in practice, it is not possible to optimise the value for α on the test set, and this would need to be done on a held-out set.) But even with this variation, similarity-class is still out-performing MDL and Assoc across the whole range of α values. The

α value	% correct – G^2	% correct – X^2
0.0005	73.8 (3.3)	74.1 (3.0)
0.05	73.4 (2.8)	73.8 (2.5)
0.3	73.0 (2.4)	74.1 (2.2)
0.75	73.9 (1.9)	74.3 (1.8)
0.995	73.8 (1.2)	73.3 (1.2)

Table 6.8: Disambiguation results for G^2 and X^2

important feature of these results is that the α values corresponding to the lowest scores lead to a significant amount of generalisation. This explains why the α value had less impact when using the complete data: the large amount of data meant that, even for low values of α, the average level of generalisation was still quite low. The results also provide additional evidence that MDL and Assoc are over-generalising for this task.

As a final experiment, we compared the task performance when using the X^2, rather than G^2, statistic in the chi-squared test. The results are given in Table 6.8 for the complete data set.[5]. The figures in brackets give the average number of generalised levels. The X^2 statistic is performing at least as well as G^2, throwing further doubt on the claim by Dunning (1993) that the G^2 statistic is better suited for use in corpus-based linguistics. Dunning's analysis was shown to be in doubt in Chapter 3. A possible explanation for the results presented here, and those in Dunning 1993, is that the X^2 statistic provides a less conservative test when counts in the contingency table are small, as is often the case in corpus-based linguistics. The pseudo disambiguation task is better served by a less conservative test, since this results in a low level of generalisation, on the whole, which is good for this task.

[5] X^2 performed slightly better than G^2 using the smaller data set also.

Chapter 7

Conclusion

This Chapter considers each of the problems that have been addressed in this thesis, outlining the proposed solution for each problem, together with the original contribution. The ways in which the work could be extended are also considered. The discussion is organised by chapter.

Chapter 3 considered the problem of how to estimate the probability of a noun sense, given a predicate and argument position. The proposed solution answers two questions: one, how to use a class from WordNet to estimate the probability of a noun sense (thereby overcoming the sparse data problem); and, two, how to select a suitable class to represent a sense. The second question can be thought of as how to select a suitable level of generalisation in WordNet. The proposed generalisation procedure employs a chi-squared test, and the level of significance of the test, α, is treated as a parameter to be set empirically. Results were given showing how the chosen level of generalisation depends on both the sample size and the value of α.

The generalisation procedure is arguably the most important contribution of the thesis. As Resnik (1993a) comments, “It has been widely noted that the selection of an appropriate level of abstraction is a difficult problem”. (p. 133) We have tried to devise a procedure that has a clearer statistical interpretation than that of Resnik, and also one that overcomes some of the shortcomings of Li and Abe’s approach, such as the uniform distribution assumption (2.13). An advantage of our approach is that treating α as a parameter gives the procedure a level of flexibility, since α can be set to produce a level of generalisation that is appropriate for the task in hand.

An alternative to using a single class to estimate the probability of a concept, which was suggested by Jason Eisner at COLING 2000, is to use all the classes dominated by the hypernyms of a concept. An estimate would be obtained for each hypernym, and the estimates combined in a linear interpolation. An approach similar to this is taken by Bikel (2000), in the context of statistical parsing.

Chapter 4 described an unsupervised reestimation algorithm for estimating sense frequencies. We first explained how splitting the count for a noun equally among its senses works better than might be expected (at least for the frequencies associated with sets of senses). The reason is that counts tend to accumulate in the right places in WordNet, namely for sets of senses that are positively associated with the predicate. This accumulation effect motivated the reestimation algorithm, in which the count for a noun is split equally on the first iteration, but, on subsequent iterations, more count is given to those noun senses that belong to ‘positively associated’ sets. A feature of the algorithm is that it employs the generalisation procedure described in Chapter 3, and this led to a new interpretation of the procedure, as one that finds sets of semantically similar senses, or ‘homogeneous’ sets of senses, in the hierarchy. The results on a pseudo disambiguation task showed that the reestimation can be beneficial in some cases.

The performance of the reestimation algorithm is limited by the fact that highly accurate WSD is unlikely to be achieved using preferences alone. Other work that has attempted to use preferences for sense disambiguation has achieved little success (Resnik 1997; Carroll and McCarthy 2000). Thus one way to further this work would be to see how other knowledge sources could be used to aid the reestimation. The surrounding context of a noun is an obvious source of additional information. There also needs to be more research into using standard estimation techniques for hidden data problems (such as the EM algorithm), building on the work by Abney and Light (1999).

A further sense ambiguity that needs consideration is ambiguity of the predicate (assuming 'predicate' refers here to a word form, and not a sense). Treating predicates as word forms conflates the preferences of the various senses. For example, the preferences of the musical sense of *play* for its object differ from the preferences of the sporting sense. It may be that techniques can be developed that are able to disambiguate both the predicate and argument simultaneously, by utilising lexical sense preferences (in combination with other knowledge sources). For example, if the object of *play* is *pool*, then the verb can be used to infer the game sense of *pool*, but, working the other way, the object can be used to infer the game sense of *play*.

Chapter 5 showed how the class-based estimation techniques can be integrated into a parse selection system. A generative model of dependency structures was presented, based on an inventory of grammatical relations. As far as possible, the techniques from Chapter 3 were used to estimate the parameters, although an alternative estimation method had to be devised for non-nominal arguments. The results were a little disappointing, since the dependency model failed to outperform the structural model of Briscoe and Carroll; however, one contribution of the chapter is that it extends similar work using WordNet (Resnik 1998; Li and Abe 1998), which has only looked at particular ambiguities in isolation, rather than the complete problem of parse selection.

The negative result is of some use, since it implies that models based on preferences alone are unlikely to achieve high accuracy, and if performance on the parse selection task is to be improved, then methods need investigating that combine lexical sense preferences with other knowledge sources. One way to achieve this would be to integrate the structural model underlying the Briscoe and Carroll parser with the dependency model. This integration would bring together two orthogonal approaches, one relying on structural relations in the parse, and the other relying on lexical relations, and is likely to benefit both approaches.

We have also shown that using the class-based estimation techniques for parse selection introduces a number of problems. First, there is the problem of disambiguating the nouns in the sentence, so that the WordNet techniques can be applied. Currently, a noun is simply replaced with one of its senses (using a simple word sense disambiguation technique), and the probability of the sense is used. A more sophisticated WSD algorithm is likely to improve performance, although the problem of WSD is a difficult, open problem. An alternative is to adopt a more integrated approach that combines parsing with WSD, and we consider Bikel's (2000) recent work in this area as particularly promising.

A second problem we encountered is that of dealing with non-nominal arguments. Class-based estimation techniques have generally only been applied to arguments from a noun hierarchy, and it is an open question whether similar techniques could be applied to arguments from other hierarchies, such as the verb hierarchy in WordNet. It is not clear whether the troponym relation in the verb hierarchy will be as useful as the hyponym relation for the purposes of probability estimation.

A further problem is obtaining large amounts of accurate training data. The performance of the parse selection model is likely to have been hampered by the lack of quality data for some grammatical relations. One potential solution to this problem is to develop unsupervised techniques, since these can be applied to very large quantities of data. An interesting question is whether unsupervised techniques similar to those of Hindle and Rooth (1993) and Ratnaparkhi (1998) can be extended to relations outside of the PP-attachment problem.

Chapter 6 presented evaluations that are more focussed on the class-based estimation techniques. It was demonstrated how these can be applied to the problem of PP-attachment, extending the original method of Hindle and Rooth (1993). It was discovered that, in order to perform well, the disambiguation method requires more training data than currently exist in treebanks, but that, with appropriate amounts of data, the method is highly accurate. It was also shown that the generalisation procedure introduced in Chapter 3 outperforms a simple approach of choosing a fixed level in the hierarchy.

A further evaluation using a pseudo disambiguation task showed that our class-based estimation method outperforms two alternative approaches based on the work of Resnik (1993a) and Li and Abe (1998). It was discovered that the alternative methods appeared to be over-generalising, at least for this task. As we have argued, a useful feature of our estimation procedure is that the level of significance used in the chi-squared test, α, can be used to guard against over or under-generalisation. But even when the results did vary with α, our method was found to outperform the alternatives across the whole range of α values.

A further useful result was that the performance on the task was at least as good when using the Pearson chi-squared statistic as when using the log-likelihood chi-squared statistic. This result is at odds with the currently accepted wisdom that the log-likelihood chi-squared statistic is a better statistic for use in corpus-based NLP. We suggested an explanation for this finding which also explains the results of Dunning (1993), who initially argued for the use of the log-likelihood statistic.

An important question that has yet to be addressed in the literature is whether class-based estimation methods perform better when the classes are automatically acquired or when they are part of a man-made hierarchy. One way to investigate this would be to perform the pseudo disambiguation task, but using clustering algorithms to estimate the probabilities. Pereira et al. (1993) and Rooth et al. (1999) have already used a similar task to evaluate their clustering algorithms; the results depended on the number of clusters induced, and ranged between 75% and 80% for both approaches, compared to the 73% reported here. Unfortunately, different test and training data were used in each case, and so it is difficult to draw any conclusions from these results. A related issue is how the structure of WordNet affects the accuracy of the probability estimates. We have taken the structure of the hierarchy for granted, without any analysis, but it may be that an alternative design would be more conducive to probability estimation.

Bibliography

Abe, N., and Li, H. (1996). Learning word association norms using tree cut pair models. In *Proceedings of the 13th International Conference on Machine Learning*, pp. 3–11 San Francisco, CA.

Abney, S. (1997). Stochastic attribute-value grammars. *Computational Linguistics*, *23*(4), 597–618.

Abney, S., and Light, M. (1999). Hiding a semantic hierarchy in a Markov model. In *Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing* University of Maryland, MD.

Abney, S., Schapire, R., and Singer, Y. (1999). Boosting applied to tagging and PP attachment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 38–45 University of Maryland, MD.

Agresti, A. (1990). *Categorical Data Analysis*. Wiley, New York.

Agresti, A. (1996). *An Introduction to Categorical Data Analysis*. Wiley, New York.

Alegre, M., Sopena, J., and Lloberas, A. (1999). PP attachment: a committee machine approach. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 231–238 University of Maryland, MD.

Allen, J. (1995). *Natural Language Understanding* (2nd edition). Benjamin/Cummings, Redwood City, CA.

Alshawi, H. (1992). *The Core Language Engine*. The MIT Press, Cambridge, MA.

Altmann, G., and Steedman, M. (1988). Interaction with context during human sentence processing. *Cognition*, *30*(3), 191–238.

Bahl, L., Jelinek, F., and Mercer, R. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *5*(2), 179–190.

Bikel, D. (2000). A statistical model for parsing and word-sense disambiguation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 155–163 Hong Kong.

Black, E., Jelinek, F., Lafferty, J., Magerman, D., Mercer, R., and Roukos, S. (1993). Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 31–37 Columbus, OH.

Bod, R. (1998). *Beyond Grammar: An Experience-Based Theory of Language*. CSLI Publications, Stanford, CA.

Boguraev, B., Briscoe, E., Carroll, J., Carter, D., and Grover, C. (1987). The derivation of a grammatically-indexed lexicon from the Longman Dictionary of Contemporary English. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pp. 193–200 Stanford, CA.

Boguraev, B., and Pustejovsky, J. (Eds.). (1996). *Corpus Processing for Lexical Acquisition*. The MIT Press, Cambridge, MA.

Bresnan, J., and Kaplan, R. (1982). Lexical-functional grammar: A formal system for grammatical representation. In Bresnan, J. (Ed.), *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge, MA.

Brill, E. (1993). Automatic grammar induction and parsing free text: A transformation-based approach. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 259–265 Columbus, OH.

Brill, E., and Resnik, P. (1994). A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 1198–1204 Kyoto, Japan.

Briscoe, E., and Carroll, J. (1993). Generalised probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, *19*(1), 25–60.

Briscoe, E., and Carroll, J. (1995). Developing and evaluating a probabilistic LR parser of part-of-speech and punctuation labels. In *Proceedings of the 4th International Workshop on Parsing Technologies*, pp. 48–58 Prague, Czech Republic.

Briscoe, E., and Carroll, J. (1997). Automatic extraction of subcategorisation from corpora. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing*, pp. 356–363 Washington, DC.

Brown, P., Della Pietra, V., deSouza, P., Lai, J., and Mercer, R. (1992). Class-based *n*-gram models of natural language. *Computational Linguistics*, *18*(4), 467–479.

Cardie, C. (1992). Corpus-based acquisition of relative pronoun disambiguation heuristics. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pp. 216–223 Newark, DE.

Carroll, J., Briscoe, E., and Sanfilippo, A. (1998a). Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, pp. 447–454 Granada, Spain.

Carroll, J., Minnen, G., and Briscoe, E. (1998b). Can subcategorisation probabilities help a statistical parser?. In *Proceedings of the 6th SIGDAT Workshop on Very Large Corpora*, pp. 118–126 Montreal, Canada.

Carroll, J., Minnen, G., and Briscoe, E. (1999). Corpus annotation for parser evaluation. In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora*, pp. 35–41 Bergen, Norway.

Carroll, J., and Briscoe, E. (1996). Apportioning development effort in a probabilistic LR parsing system through evaluation. In *Proceedings of the SIGDAT Conference on Empirical Methods in Natural Language Processing*, pp. 92–100 Philadelphia, PA.

Carroll, J., and McCarthy, D. (2000). Word sense disambiguation using automatically acquired verbal preferences. *Computers and the Humanities*, *34*(1-2), 109–114.

Charniak, E. (1993). *Statistical Language Learning*. The MIT Press, Cambridge, MA.

Charniak, E. (1997). Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pp. 598–603 Menlo Park, CA.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 132–139 Seattle, WA.

Chen, S., and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pp. 310–318 Santa Cruz, CA.

Chiang, D. (2000). Statistical parsing with an automatically-extracted tree adjoining grammar. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 456–463 Hong Kong.

Church, K. (1988). A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the 2nd ACL Conference on Applied Natural Language Processing*, pp. 136–143 Austin, TX.

Church, K., and Gale, W. (1991). A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language*, *5*, 19–54.

Church, K., Gale, W., Hanks, P., and Hindle, D. (1991). Using statistics in lexical analysis. In Zernik, U. (Ed.), *Lexical Acquisition: Using On-line Resources to Build a Lexicon*, chap. 6, pp. 115–164. Lawrence Erlbaum, Hillsdale, NJ.

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

Church, K., and Patil, R. (1982). Coping with syntactic ambiguity or how to put the block in the box on the table. *American Journal of Computational Linguistics*, *8*(3-4), 139–149.

Ciaramita, M., and Johnson, M. (2000). Explaining away ambiguity: Learning verb selectional preference with Bayesian networks. In *Proceedings of the 18th International Conference on Computational Linguistics*, pp. 187–193 Saarbrucken, Germany.

Clark, S., and Weir, D. (1999). An iterative approach to estimating frequencies over a semantic hierarchy. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 258–265 University of Maryland, MD.

Clark, S., and Weir, D. (2000). A class-based probabilistic approach to structural disambiguation. In *Proceedings of the 18th International Conference on Computational Linguistics*, pp. 194–200 Saarbrucken, Germany.

Clark, S., and Weir, D. (2001). Class-based probability estimation using a semantic hierarchy. In *Proceedings of the 2nd Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 95–102 Pittsburgh, PA.

Collins, M. (1996). A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pp. 184–191 Santa Cruz, CA.

Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pp. 16–23 Madrid, Spain.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Collins, M., and Brooks, J. (1995). Prepositional phrase attachment through a backed-off model. In *Proceedings of the 3rd SIGDAT Workshop on Very Large Corpora*, pp. 27–38 Cambridge, MA.

Cover, T., and Thomas, J. (1991). *Elements of Information Theory*. Wiley, New York.

Daelemans, W., Van Den Bosch, A., and Zavrel, Z. (1999). Forgetting exceptions is harmful in language learning. *Machine Learning*, *34*(1-3), 11–43.

Dagan, I., Lee, L., and Pereira, F. (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning*, *34*(1-3), 43–69.

Dagan, I., Marcus, S., and Markovitch, S. (1995). Contextual word similarity and estimation from sparse data. *Computer Speech and Language*, *9*, 123–152.

Drange, T. (1966). *Type Crossings: Sentential Meaningless in the Border Area of Linguistics and Philosophy*. Mouton, The Hague.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, *19*(1), 61–74.

Eisner, J. (1996a). An empirical comparison of probability models for dependency grammar. Tech. rep. IRCS-96-11, University of Pennsylvania.

Eisner, J. (1996b). Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics*, pp. 340–345 Copenhagen, Denmark.

Fellbaum, C. (1998a). A semantic network of English verbs. In Fellbaum, C. (Ed.), *WordNet: An Electronic Lexical Database*, chap. 3, pp. 69–104. The MIT Press, Cambridge, MA.

Fellbaum, C. (Ed.). (1998b). *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA.

Fisher, D., and Riloff, E. (1992). Applying statistical methods to small corpora: Benefiting from a limited domain. In *AAAI Symposium on Probabilistic Approaches to Natural Language*, pp. 47–53 Cambridge, MA.

Fodor, J. (1977). *Semantics: theories of meaning in generative grammar*. Harvard University Press.

Franz, A. (1996). *Automatic Ambiguity Resolution in Natural Language Processing*. Springer, Berlin, Germany.

Frazier, L. (1978). *On Comprehending Sentences: Syntactic Parsing Strategies*. Ph.D. thesis, University of Connecticut.

Gale, W., and Church, K. (1990). Poor estimates of context are worse than none. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pp. 283–287 Hidden Valley, PA.

Gale, W., and Church, K. (1994). What's wrong with adding one?. In Oostdijk, N., and de Haan, P. (Eds.), *Corpus-Based Research into Language*. Rodolpi, Amsterdam, The Netherlands.

Gale, W., and Sampson, G. (1995). Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics*, *2*(3), 217–237.

Ge, N., Hale, J., and Charniak, E. (1998). A statistical approach to anaphora resolution. In *Proceedings of the 6th SIGDAT Workshop on Very Large Corpora*, pp. 161–170 Montreal, Canada.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, *40*(3–4), 237–264.

Goodman, J. (1997). Probabilistic feature grammars. In *Proceedings of the 5th International Workshop on Parsing Technologies*, pp. 89–100 Boston, MA.

Grishman, R., Macleod, C., and Meyers, A. (1994). Comlex syntax: Building a computational lexicon. In *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 268–272 Kyoto, Japan.

Harrison, P., Abney, S., Black, E., Flickinger, D., Gdaniec, C., Grishman, R., Hindle, D., Ingria, B., Marcus, M., Santorini, B., and Strzalkowski, T. (1991). Evaluating syntax performance of parser/grammars of English. In *Proceedings of the Workshop on Evaluating Natural Language Processing Systems*, pp. 71–77 Griffiss Air Force Base, NY.

Hektoen, E. (1997). Probabilistic parse selection based on semantic cooccurrences. In *Proceedings of the 5th International Workshop on Parsing Technologies*, pp. 113–122 Boston, MA.

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

Hindle, D. (1994). A parser for text corpora. In Atkins, B., and Zampolli, A. (Eds.), *Computational Approaches to the Lexicon*, pp. 103–151. Oxford University Press, Oxford, UK.

Hindle, D., and Rooth, M. (1991). Structural ambiguity and lexical relations. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp. 229–236 Berkeley, CA.

Hindle, D., and Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics*, *19*(1), 103–120.

Hobbs, J., and Bear, J. (1990). Two principles of parse preference. In *Proceedings of the 13th International Conference on Computational Linguistics*, pp. 162–167 Helsinki, Finland.

Hockenmaier, J., Bierner, G., and Baldridge, J. (2000). Providing robustness for a CCG system. In *Proceedings of the ESSLLI-2000 Workshop on Linguistic Theory and Grammar Implementation* Birmingham, UK.

Hofland, K., and Johansson, S. (1982). *Word Frequencies in British and American English.* Norwegian Computing Centre for the Humanities, Bergen, Norway.

Jelinek, F., Lafferty, J., Magerman, D., Mercer, R., Ratnaparkhi, A., and Roukos, S. (1994). Decision tree parsing using a hidden derivation model. In *Proceedings of the ARPA Human Language Technology Workshop*, pp. 272–277 Plainsboro, NJ.

Jelinek, F., and Mercer, R. (1980). Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pp. 381–397 Amsterdam, The Netherlands.

Jelinek, F. (1997). *Statistical Methods for Speech Recognition.* The MIT Press, Cambridge, MA.

Johnson, D., and Postal, P. (1980). *Arc Pair Grammar.* Princeton University Press, Princeton, NJ.

Johnson, M., Geman, S., Canon, S., Chi, S., and Riezler, S. (1999). Estimators for stochastic ‘unification-based’ grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 535–541 University of Maryland, MD.

Joshi, A., and Schabes, Y. (1992). Tree-adjoining grammars and lexicalized grammars. In Nivat, M., and Podelski, A. (Eds.), *Definability and Recognizability of Sets of Trees*. Elsevier, Princeton, NJ.

Katz, J., and Fodor, J. (1964). The structure of a semantic theory. In Fodor, J., and Katz, J. (Eds.), *The Structure of Language*, chap. 19, pp. 479–518. Prentice-Hall, Englewood Cliffs, NJ.

Katz, S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *35*(3), 400–401.

Kilgarriff, A. (1996). Which words are particularly characteristic of a text? A survey of statistical approaches. In *Proceedings of the AISB Workshop on Language Engineering for Document Analysis and Recognition*, pp. 33–40 Sussex, UK.

Kilgarriff, A., and Rose, T. (1998). Measures for corpus similarity and homogeneity. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing*, pp. 46–52 Granada, Spain.

Kimball, J. (1973). Seven principles of surface structure parsing in natural language. *Cognition*, *2*, 15–47.

Lafferty, J., Sleator, D., and Temperley, D. (1992). Grammatical trigrams: A probabilistic model of link grammar. In *AAAI Symposium on Probabilistic Approaches to Natural Language*, pp. 89–97 Cambridge, MA.

Larson, H. J. (1982). *Introduction to Probability Theory and Statistical Inference* (3rd edition). Wiley, Singapore.

Lauer, M. (1995). Corpus statistics meet the compound noun: some empirical results. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 47–55 Cambridge, MA.

Lee, L. (1999). Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 25–32 University of Maryland, MD.

Lee, L., and Pereira, F. (1999). Distributional similarity models: Clustering vs. nearest neighbors. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 33–40 University of Maryland, MD.

Li, H. (1996). A probabilistic disambiguation method based on psycholinguistic principles. In *Proceedings of the 4th SIGDAT Workshop on Very Large Corpora*, pp. 141–154 Copenhagen, Denmark.

Li, H. (1998). *A Probabilistic Approach to Lexical Semantic Knowledge Acquisition and Structural Disambiguation.* Ph.D. thesis, University of Tokyo.

Li, H., and Abe, N. (1996). Clustering words with the MDL principle. In *Proceedings of the 16th International Conference on Computational Linguistics*, pp. 4–9 Copenhagen, Denmark.

Li, H., and Abe, N. (1998). Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics*, *24*(2), 217–244.

Li, H., and Abe, N. (1999). Learning dependencies between case frame slots. *Computational Linguistics*, *25*(2), 283–291.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pp. 768–773 Montreal, Canada.

Magerman, D. (1994). *Natural Language Parsing as Statistical Pattern Recognition.* Ph.D. thesis, University of Stanford.

Magerman, D. (1995). Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 276–283 Cambridge, MA.

Magerman, D., and Marcus, M. (1990). Parsing a natural language using mutual information statistics. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pp. 984–989 Boston, MA.

Manning, C., and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA.

Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330.

McCarthy, D. (1997). Word sense disambiguation for acquisition of selectional preferences. In *Proceedings of the ACL/EACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pp. 52–61 Madrid, Spain.

McCarthy, D. (2000). Using semantic preferences to identify verbal participation in role switching. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 256–263 Seattle, WA.

McCarthy, D., and Korhonen, A. (1998). Detecting verbal participation in diathesis alternations. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pp. 1493–1495 Montreal, Canada.

McCawley, J. (1968). The role of semantics in a grammar. In Bach, E., and Harms, R. (Eds.), *Universals in Linguistic Theory*, pp. 125–169. Holt, Rinehart and Winston, New York.

Melcuk, I. (1988). *Dependency Syntax: Theory and Practice.* State University of New York Press.

Miller, G., Leacock, C., Tengi, R., and Bunker, R. (1993). A semantic concordance. In *Proceedings of the ARPA Human Language Technology Workshop*, pp. 303–308 Plainsboro, NJ.

Miller, G. (1998). Nouns in WordNet. In Fellbaum, C. (Ed.), *WordNet: An Electronic Lexical Database*, chap. 1, pp. 23–46. The MIT Press, Cambridge, MA.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA.

Pedersen, T. (1996). Fishing for exactness. In *Proceedings of the South–Central SAS Users Group Conference* Austin, TX.

Pereira, F., Tishby, N., and Lee, L. (1993). Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 183–190 Columbus, OH.

Pollard, C., and Sag, I. (1994). *Head-Driven Phrase Structure Grammar.* The University of Chicago Press, Chicago, IL.

Prescher, D., Riezler, S., and Rooth, M. (2000). Using a probabilistic class-based lexicon for lexical ambiguity resolution. In *Proceedings of the 18th International Conference on Computational Linguistics*, pp. 649–655 Saarbrucken, Germany.

Ratnaparkhi, A. (1997). A linear observed time statistical parser based on maximum entropy models. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, pp. 1–10 Brown University, Providence, RI.

Ratnaparkhi, A. (1998). Unsupervised statistical models for prepositional phrase attachment. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pp. 1079–1085 Montreal, Canada.

Ratnaparkhi, A. (1999). Learning to parse natural language with maximum entropy models. *Machine Learning*, *34*(1-3), 151–175.

Ratnaparkhi, A., Reynar, J., and Roukos, S. (1994). A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Human Language Technology Workshop*, pp. 250–255 Plainsboro, NJ.

Resnik, P. (1993a). *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph.D. thesis, University of Pennsylvania.

Resnik, P. (1993b). Semantic classes and syntactic ambiguity. In *Proceedings of the ARPA Human Language Technology Workshop*, pp. 278– 283 Princeton, NJ.

Resnik, P. (1996). Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, *61*, 127–159.

Resnik, P. (1997). Selectional preference and sense disambiguation. In *Proceedings of the 1997 SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, pp. 52–57 Washington, DC.

Resnik, P. (1998). WordNet and class-based probabilities. In Fellbaum, C. (Ed.), *WordNet: An Electronic Lexical Database*, chap. 10, pp. 239–263. The MIT Press, Cambridge, MA.

Resnik, P. (1999a). Disambiguating noun groupings with respect to WordNet senses. In Armstrong, S., Church, K., Isabelle, P., Manzi, S., Tzoukermann, E., and Yarowsky, D. (Eds.), *Natural Language Processing Using Very Large Corpora*, pp. 77–98. Kluwer Academic Publishers.

Resnik, P. (1999b). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, *11*, 95–130.

Resnik, P., and Hearst, M. (1993). Syntactic ambiguity and conceptual relations. In *Proceedings of the ACL Workshop on Very Large Corpora*, pp. 58–64 Columbus, OH.

Resnik, P., and Yarowsky, D. (1997). A perspective on word sense disambiguation methods and their evaluation. In *Proceedings of the 1997 SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, pp. 79–86 Washington, DC.

Ribas, F. (1994). An experiment on learning appropriate selectional restrictions from a parsed corpus. In *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 769–774 Kyoto, Japan.

Ribas, F. (1995a). *On Acquiring Appropriate Selectional Restrictions from Corpora Using a Semantic Taxonomy*. Ph.D. thesis, Technical University of Catalonia.

Ribas, F. (1995b). On learning more appropriate selectional restrictions. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 112–118 Dublin, Ireland.

Riezler, S., Prescher, D., Kuhn, J., and Johnson, M. (2000). Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 480–487 Hong Kong.

Rooth, M., Riezler, S., Prescher, D., Carroll, G., and Beil, F. (1999). Inducing a semantically annotated lexicon via EM-based clustering. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 104–111 University of Maryland, MD.

Sampson, G. (1995). *English for the Computer.* Oxford University Press, Oxford, UK.

Steedman, M. (2000). *The Syntactic Process*. The MIT Press, Cambridge, MA.

Stetina, J., and Nagao, M. (1997). Corpus based PP attachment ambiguity resolution with a semantic dictionary. In *Proceedings of the 5th SIGDAT Workshop on Very Large Corpora*, pp. 66–80 Beijing, China.

Taraban, R., and McClelland, J. (1988). Constituent attachment and thematic role assignment in sentence processing: Influences of content-based expectations. *Journal of Memory and Language*, *27*, 597–632.

Wagner, A. (2000). Enriching a lexical semantic net with selectional preferences by means of statistical corpus analysis. In *Proceedings of the ECAI-2000 Workshop on Ontology Learning* Berlin, Germany.

Whittemore, G., Ferrara, K., and Brunner, H. (1990). Empirical study of predictive powers of simple attachment schemes for post-modifier prepositional phrases. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pp. 23–30 Pittsburgh, PA.

Wilks, Y. (1975). An intelligent analyzer and understander of English. *Communications of the ACM*, *18*(5), 264–274.

Wilks, Y., and Fass, D. (1992). The preference semantics family. *Computers Mathematics Applications*, *23*(2-5), 205–221.

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*, pp. 454–460 Nantes, France.

Yeh, A. (2000a). Comparing two trainable grammatical relations finders. In *Proceedings of the 18th International Conference on Computational Linguistics*, pp. 1146–1150 Saarbrucken, Germany.

Yeh, A. (2000b). Using existing systems to supplement small amounts of annotated grammatical relations training data. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 126–132 Hong Kong.

Zavrel, J., Daelemans, W., and Veenstra, J. (1997). Resolving PP attachment ambiguities with memory-based learning. In *Proceedings of the ACL workshop on Computational Natural Language Learning*, pp. 136–144 Madrid, Spain.

Appendix A

Grammatical Relations used in the Implementation of the Parse Selection System

Some of the descriptions given here are taken directly from Carroll et al. 1998a, and the same notation is used. Many of the examples also come directly from that paper.[1]

mod(type,head,dependent) The relation between a head and a modifier; **type** is used to indicate the word introducing the dependent (where appropriate). Examples include the following:

mod(_,flag,red)	a red flag
mod(with,walk,John)	walk with John
mod(while,walk,talk)	walk while talking
mod(_,Picasso,painter)	Picasso the painter
mod(of,examination,patient)	the examination of the patient
mod('s,doctor,examination)	the doctor's examination

Clausal and non-clausal modifiers may be distinguished by the use of **cmod**, **xmod** and **ncmod**; **cmod** is for clausal modifiers controlled from within, **xmod** is for clausal modifiers controlled from without. Examples include the following:

ncmod(in,meeting,London)	the meeting in London
ncmod(_,burden,disproportionate)	a disproportionate burden
xmod(without,eat,ask)	he ate the cake without asking
cmod(because,eat,be)	he ate the cake because he was hungry

detmod(type,head,dependent) The relation between a noun and a determiner; **type** is usually empty, but has the value **poss** for pronominal determiners; for example:

detmod(_,burden,a)	a burden
detmod(_,meeting,the)	the meeting
detmod(poss,system,his)	his system

arg_mod(type,head,dependent,initial_gr) The relation between a head and a semantic argument which is syntactically realised as a modifier. The **type** slot indicates the word introducing the dependent, and **initial_gr** is used to indicate the grammatical relation before any transformation; for example:

arg_mod(by,kill,Brutus,subj)	killed by Brutus
arg_mod(by,acquire,proprietor,subj)	acquired by proprietor

[1] Thanks to John Carroll for giving permission to use the descriptions and examples.

ncsubj(head,dependent,initial_gr) The relation between a predicate and a non-clausal subject; where appropriate, **initial_gr** is obj after passivisation; for example:

ncsubj(arrive,John,_)	John arrived in Paris
ncsubj(employ,Microsoft,_)	Microsoft employed 10 C programmers
ncsubj(employ,Paul,obj)	Paul was employed by IBM

c/xsubj(head,dependent,initial_gr) The relation between a predicate and a clausal subject, controlled from within, and from without, respectively; for example:

csubj(mean,leave,_)	that Nellie left without saying good-bye meant she was angry
csubj(astonish,owe,_)	that he owed anything would have astonished his mother
xsubj(require,win,_)	to win the America's Cup requires heaps of cash

dobj(head,dependent,initial_gr) The relation between a predicate and a direct object; where appropriate, **initial_gr** is **iobj** after dative shift; e.g.

dobj(read,book,_)	read books
dobj(mail,Mary,iobj)	mail Mary the contract

obj2(head,dependent) The relation between a predicate and the second non-clausal complement in ditransitive constructions; for example:

obj2(give,present)	give Mary a present
obj2(mail,contract)	mail Mary the contract

iobj(type,head,dependent) The relation between a predicate and a non-clausal complement introduced by a preposition; **type** indicates the preposition; for example:

iobj(in,arrive,Spain)	arrive in Spain
iobj(into,put,box)	put into box
iobj(to,give,poor)	give to the poor

clausal(type,head,dependent) The relation between a predicate and a clausal complement; **type** indicates the complementizer or preposition introducing the complement, where appropriate.

c/xcomp(type,head,dependent) The relation between a predicate and a clausal complement with and without an overt subject, respectively; for example:

xcomp(to,intend,leave)	Paul intends to leave IBM
xcomp(_,be,easy)	Swimming is easy
xcomp(in,be,Paris)	Mary is in Paris
xcomp(_,be,manager)	Paul is the manager
ccomp(that,say,accept)	Paul said that he will accept Microsoft's offer
ccomp(that,say,leave)	I said that he left

aux(head,dependent) The relation between an auxiliary verb and a main or other auxiliary verb; for example:

aux(continue,will)	this will continue to place a burden on the tax-payer
aux(be,should)	the final should be a good game