Measures and Applications of Lexical Distributional Similarity

Julie Elizabeth Weeds

Submitted for the degree of D. Phil.

University of Sussex

30th September 2003

Declaration

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other university for a degree.

Signature:

For my Mum

© Julie Elizabeth Weeds, September 2003

Measures and Applications of Lexical Distributional Similarity

Julie Elizabeth Weeds

Summary

This thesis is concerned with the measurement and application of lexical distributional similarity. Two words are said to be distributionally similar if they appear in similar contexts. This loose definition, however, has led to many measures being proposed or adopted from fields such as geometry, statistics, Information Retrieval (IR) and Information Theory. Our aim is to investigate the properties which make a good measure of lexical distributional similarity. We start by introducing the concept of lexical distributional similarity. We discuss potential applications, which can be roughly divided into distributional or language modelling applications and semantic applications, and methods of evaluation (Chapter 2). We look at existing measures of distributional similarity and carry out an empirical comparison of fifteen of these measures, paying particular attention to the effects of word frequency (Chapter 3). We propose a new general framework for distributional similarity based on the concept of lexical substitutability, which we measure using the IR concepts of precision and recall. This framework allows us to investigate the key factors in similarity of asymmetry, the relative influence of different contexts and the extent to which words share a context (Chapter 4). Finally, we consider the application of distributional similarity in language modelling (Chapter 5) and as a predictor of semantic similarity using human judgements of similarity and a spelling correction task (Chapter 6).

Submitted for the degree of D. Phil.

University of Sussex

30th September 2003

Acknowledgements

Four years ago, many of my friends and colleagues were surprised when I quit my city job after just one year to start a career in research. For me, the decision was easy. I wanted to spend my life doing things I enjoyed. Four years on, it's a decision I'm still happy with.

I realise how lucky I am to be able to say that I have thoroughly enjoyed my DPhil experience. This is of course largely due to the many people who have had an influence on my life and studies over the course of the last three years. First and foremost, I thank my supervisor, David Weir, for his guidance and inspiration. Our weekly meetings have undoubtedly been key in keeping me positive when otherwise I might have begun to doubt myself. Second, without a grant from the Engineering and Physical Sciences Research Council, none of this work would have been possible.

I would also like to thank a number of other NLP people, variously at Sussex, Brighton and Edinburgh: my degree committee, Adam Kilgarriff and Bill Keller, for their thoughtful input; John Carroll, for providing the raw data for my experiments; Darren Pearce, a fellow student, friend and LaTeX guru, for all of his support; and Mark McLauchlan, Diana McCarthy, John Carroll, Rudi Lutz, Chong-hoon Ha, Stephen Clark and James Curran for helpful discussions. Finally, it would not have been possible without the support of my family and friends. In particular, I thank my Mum, Pauline Weeds, who has always supported me in all my choices. I also thank the other members of my family, my friends (in Brighton, Medway, Cambridge and beyond) and the University of Sussex rowing club, whose combined ability to distract me has probably kept me sane.

Bibliographic Notes

Portions of the research on which this thesis is based have already been published in the form of conference papers, co-written with David Weir. An initial form of the co-occurrence retrieval framework, described in Chapter 4, was presented at the 2003 Conference on Empirical Methods in Natural Language Processing (Weeds & Weir, 2003b). An evaluation of two existing distrbutional similarity measures for use in distributional and semantic applications was presented at the Second International Conference on Corpus Linguistics (Weeds & Weir, 2003a). The work on smoothing in Section 5.1 is also described in Weeds (2003) and preliminary work on pseudo-disambiguation experiments (Chapters 3 and 5) is described in Weeds (2002).

Contents

1 Introduction **13**

2 Background **17**

2.1 Lexical Distributional Similarity 17

2.2 Applications 22

2.2.1 Classification of Applications 22

2.2.2 Language Modelling 24

2.2.3 Prepositional Phrase Attachment Ambiguity Resolution 25

2.2.4 Conjunction Scope Identification 26

2.2.5 Word Sense Disambiguation 27

2.2.6 Word Sense Separation 28

2.2.7 Metonymy Resolution 29

2.2.8 Learning Countability Preferences 29

2.2.9 Automatic Thesaurus Generation 30

2.2.10 Information Retrieval 31

2.2.11 Anaphor Resolution 31

2.2.12 Collocation Extraction 32

2.2.13 Text Simplification 33

2.2.14 Compound Noun Interpretation 34

2.2.15 Topic Identification 34

2.2.16 Spelling Correction 34

2.2.17 Clustering 35

2.3 Evaluation of Distributional Similarity Measures 36

2.3.1 Evaluation Against a Gold Standard 36

2.3.2 Evaluation Against Human Plausibility Judgements 38

2.3.3 Evaluation Against Unseen Data 39

2.3.4 Application-Based Evaluation Tasks 40

2.3.5 Pseudo-Disambiguation Tasks 41

2.3.6 WordNet Prediction Task 43

2.4 Conclusions 44

3 Existing Measures of Distributional Similarity **46**

3.1 Theory 46

3.1.1 Geometric Measures 47

3.1.2 Correlation Coefficients 50

3.1.3 Combinatorial Measures 52

3.1.4 Substitutability Measures 54

3.1.5 Mutual Information Based Measures . . . 57
3.2 Empirical Comparison of Measures . . . 59
3.2.1 Data . . . 60
3.2.2 Cross-Comparison of Neighbours Derived Using Different Measures . . 65
3.2.3 High or Low Frequency Neighbours? . . . 67
3.3 Conclusions . . . 74

4 A General Framework for Distributional Similarity 75
4.1 Co-occurrence Retrieval . . . 75
4.1.1 Additive Models . . . 76
4.1.2 Difference-Weighted Models . . . 79
4.1.3 Combining Precision and Recall . . . 82
4.1.4 Conclusion . . . 83
4.2 Neighbours Selected by Co-occurrence Retrieval Models . . . 84
4.2.1 Neighbours of "hope" . . . 84
4.2.2 Relationship between Parameter Settings and Frequency of Neighbours Selected . . . 84
4.2.3 Conclusion . . . 91
4.3 Characterisations of Existing Measures . . . 92
4.3.1 The L_1 Norm . . . 93
4.3.2 The L_2 Norm . . . 95
4.3.3 Cosine Measure . . . 95
4.3.4 Kendall's τ Coefficient . . . 96
4.3.5 Dice and Jaccard's Coefficient . . . 97
4.3.6 Jensen-Shannon Divergence Measure . . . 99
4.3.7 The α-skew Divergence Measure . . . 100
4.3.8 Confusion Probability . . . 102
4.3.9 Hindle's Measure . . . 106
4.3.10 Lin's MI-Based Measure . . . 106
4.3.11 MI Variant of Jaccard's Coefficient . . . 109
4.4 Conclusion . . . 109

5 Language Modelling 112
5.1 Deriving Probability Distributions . . . 112
5.1.1 Introduction . . . 112
5.1.2 Smoothing Algorithm . . . 114
5.1.3 Evaluation . . . 117
5.1.4 Results . . . 118
5.1.5 Conclusion . . . 121
5.2 Pseudo-Disambiguation Experiments . . . 122
5.2.1 Experimental Set-up . . . 122
5.2.2 Results . . . 127
5.2.3 Correlation between Pseudo-Disambiguation Results and Smoothing Results 136

5.2.4 Discussion . . . 138
5.3 Conclusions . . . 141

6 Predicting Semantic Similarity 142
6.1 Correlation with Human Synonymy Judgements . . . 142
6.1.1 Methodology . . . 142
6.1.2 Results . . . 143
6.1.3 Discussion . . . 148
6.2 Comparison with WordNet . . . 151
6.2.1 Neighbour Set Comparisons . . . 152
6.2.2 Hyponyms vs. Hyperonyms . . . 155
6.3 Spelling Correction Task . . . 162
6.3.1 Motivation . . . 163
6.3.2 Methodology . . . 163
6.3.3 Results . . . 165
6.3.4 Discussion . . . 167
6.4 Conclusions . . . 169

7 Conclusions and Future Directions 172
7.1 Summary of Conclusions . . . 172
7.1.1 Chapter 1 . . . 172
7.1.2 Chapter 2 . . . 172
7.1.3 Chapter 3 . . . 173
7.1.4 Chapter 4 . . . 173
7.1.5 Chapter 5 . . . 174
7.1.6 Chapter 6 . . . 175
7.2 Major Contributions . . . 177
7.3 Major Directions for Further Work . . . 178

References 180

Index 192

A Sets of Nouns Used in Experiments 194
A.1 High Frequency Nouns . . . 194
A.2 Low Frequency Nouns . . . 198

List of Figures

3.1 2-d vector diagram for *bus* and *dog* 47
3.2 Geometric distance metrics 48
3.3 The cosine measure 49
3.4 Positive correlation between bus and train 51
3.5 Correlation between bus and dog 51
3.6 Venn Diagram for similarity between n_1 and n_2 53
3.7 Performance of Neighbour Sets on Different Test Sets 63

4.1 Tendencies of additive type-based CRM to select high frequency neighbours for high and low frequency target nouns 89
4.2 Tendencies for additive token-based CRM to select high frequency neighbours for high and low frequency target nouns 90
4.3 Tendencies for additive MI-based CRM to select high frequency neighbours for high and low frequency target nouns 91
4.4 Tendencies for difference-weighted type-based CRM to select high frequency neighbours for high and low frequency target nouns 92
4.5 Tendencies for difference-weighted MI-based model to select high frequency neighbours for high and low frequency target nouns 93
4.6 Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive type-based CRM and of sim_{τ} 98
4.7 Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive type-based CRM and of sim_{jacc}. 99
4.8 Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive token-based CRM and of $dist_{\alpha 1}$ 102
4.9 Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive token-based CRM and of $dist_{\alpha 2}$ 103
4.10 Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive token-based CRM and of sim_{cp} 104
4.11 Variation (with parameters β and γ) in mean similarity between neighbour sets of the difference-weighted MI-based CRM and of $sim_{cp_{rev}}$ 105
4.12 Variation (with parameters β and γ) in mean similarity between neighbour sets of the difference-weighted MI-based CRM and of sim_{hind+} 107
4.13 Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive MI-based CRM and of sim_{lin} 108
4.14 Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive MI-based CRM and $sim_{jacc_{mi}}$ 110

5.1 Algorithm for smoothing co-occurrence distribution of target word using nearest neighbours . . . 115
5.2 Distance between half the data set and the whole data set plotted against word frequency . . . 117
5.3 Results for stopping condition 3 in the language modelling task . . . 120
5.4 Comparison of error rates between one vote per neighbour and frequency weighted voting schemes . . . 125
5.5 Barchart illustrating cross-validated optimal error rates for each measure when k is optimised . . . 129
5.6 Performance of existing similarity measures with respect to k . . . 131
5.7 Performance of CRMs with respect to k (at optimal values of β and γ) . . . 132
5.8 Performance of CRMs with respect to β (at optimal values of k and γ) . . . 133
5.9 Performance of CRMs with respect to γ (at optimal values of k and β) . . . 134
5.10 Barchart illustrating cross-validated optimal error rates for each measure when t is optimised . . . 135
5.11 Performance of existing similarity measures with respect to t . . . 137
5.12 Performance of CRMs with respect to t (at optimal values of β and γ) . . . 138
5.13 Performance of CRMs with respect to β (at optimal values of t and γ) . . . 139
5.14 Performance of CRMs with respect to γ (at optimal values of t and β) . . . 140

6.1 Correlation with M&C data for the additive type-based CRM . . . 146
6.2 Correlation with M&C data for additive token-based CRM . . . 146
6.3 Correlation with M&C data for additive MI-based CRM . . . 147
6.4 Correlation with M&C data for difference-weighted type-based model . . . 147
6.5 Correlation with M&C data for difference-weighted MI-based model . . . 148
6.6 Scatter Graph for WordNet-Based Measure . . . 148
6.7 Scatter Graph for Additive MI-Based Model ($\gamma = 0.75, \beta = 1.0$) . . . 149
6.8 Scatter Graph for α-skew Divergence Measure ($dist_{\alpha}(A,B)$) . . . 149
6.9 Barchart illustrating correlation with WordNet for each measure . . . 153
6.10 Variation in correlation with WordNet with respect to β and γ for the additive type-based CRM . . . 154
6.11 Variation in correlation with WordNet with respect to β and γ for the additive token-based CRM . . . 155
6.12 Variation in correlation with WordNet with respect to β and γ for the additive MI-based CRM . . . 156
6.13 Variation in correlation with WordNet with respect to β and γ for the difference-weighted type-based CRM . . . 157
6.14 Variation in correlation with WordNet with respect to β and γ for the difference-weighted MI-based CRM . . . 158
6.15 Frequency in BNC Data Against Depth in WordNet for 2000 Nouns . . . 159
6.16 Variation in F on spelling correction task with respect to parameters k and scope for sim_{wn} . . . 166

6.17 Variation in precision, recall and F as the number of nearest neighbours considered, k, is varied, for sim_{wn} at scope = 150 words 167
6.18 Variation in F on spelling correction task with respect to parameters k and scope for sim_{lin} 168
6.19 Variation in F-Score on spelling correction task with respect to parameters k and scope for the additive MI-based CRM at ($\gamma = 0, \beta = 0.9$). 169
6.20 Variation in F-Score on spelling correction task with respect to parameters β and γ for the additive MI-based CRM at $k = 1$ and scope=150. 170

List of Tables

2.1 Classification of potential applications of distributional similarity measures according to whether they primarily make use of distributional information or semantic information 23

3.1 Co-occurrence information for *bus* and *dog* 47
3.2 Sizes of intersections of feature sets 53
3.3 Fifteen similarity measures 60
3.4 Accuracy of RASP 62
3.5 Neighbours of *hope*, with their associated similarity scores, according to each similarity measure 66
3.6 Mean cross measure correlation of neighbour sets for high frequency nouns. Figures in brackets are standard deviations 68
3.7 Mean cross measure correlation of neighbour sets for low frequency nouns. Figures in brackets are standard deviations 69
3.8 Average value of $biashigh_{sim}$ for high frequency target nouns and for low frequency target nouns. Values above 0.5 indicate that the measure is biased towards selecting high frequency nouns as neighbours (for those target nouns) and values less than 0.5 indicate that the measure is biased towards selecting low frequency nouns as neighbours (for those target nouns). 71
3.9 Measures grouped by tendency to select high frequency, low frequency or similar frequency neighbours 71
3.10 Contingency table showing relationship between distribution width and frequency for 1000 most frequent nouns 72
3.11 Error rates on pseudo-disambiguation task for four measures using *true* neighbours and two restricted neighbour sets. Figures in brackets are the values of *k*, the neighbourhood size, at which the minimum error rates were found. 73
3.12 Error rates on pseudo-disambiguation task for $dist_{\alpha 1}$ and $dist_{\alpha 2}$ measures using *true* neighbours and *most frequent* neighbours. Figures in brackets are the values of *k* at which the minimum error rates were found. 73

4.1 Table of special values of β and γ 83
4.2 Neighbours of “hope” using additive type-based CRM 85
4.3 Neighbours of “hope” using additive token-based CRM 85
4.4 Neighbours of “hope” using additive MI-based CRM 86
4.5 Neighbours of “hope” using difference-weighted type-based CRM 86
4.6 Neighbours of “hope” using difference-weighted token-based CRM. The parameter settings are irrelevant in this model since precision is equal to recall. 87

4.7 Neighbours of "hope" using difference-weighted MI-based CRM 87
4.8 Summary of tendencies to select high or low frequency neighbours by models in the framework 88
4.9 Summary of observations about existing measures with reference to the CR Framework 94
4.10 Mean optimal similarities between CRMs and $dist_{L_2}$. Figures in brackets are standard deviations. 95
4.11 Mean optimal similarities between CRMs and sim_{cos}. Figures in brackets are standard deviations. 96
4.12 Mean optimal similarities between CRMs and sim_{τ}. Figures in brackets are standard deviations. 97
4.13 Mean optimal similarities between CRMs and $dist_{js}$. Figures in brackets are standard deviations. 100
4.14 Mean optimal similarities between CRMs and $dist_{\alpha 1}$. Figures in brackets are standard deviations. 101
4.15 Mean optimal similarities between CRMs and $dist_{\alpha 2}$. Figures in brackets are standard deviations. 101
4.16 Mean optimal similarities between CRMs and sim_{cp}. Figures in brackets are standard deviations. 103
4.17 Mean optimal similarities between CRMs and $sim_{cp_{rev}}$. Figures in brackets are standard deviations. 105
4.18 Mean optimal similarities between CRMs and $sim_{jacc_{mi}}$. Figures in brackets are standard deviations. 109

5.1 Comparison of stopping conditions and baselines for language modelling 119
5.2 Mean optimal error rates using five-fold cross-validation (when optimising k, γ and β). Figures in brackets are standard deviations in the optimal error rates across the five folds. 128
5.3 Summary of results on pseudo-disambiguation task when optimising k, the number of nearest neighbours 130
5.4 Mean optimal error rates using five-fold cross-validation (when optimising t, γ and β). Figures in brackets are standard deviations. 135
5.5 Summary of results of pseudo-disambiguation task when optimising t, the similarity threshold 136

6.1 Correlations between WordNet based similarity measures and M&C data according to Budanitsky and Hirst (2001) and Jarmasz and Szpakowicz (2003) 143
6.2 Comparison of existing similarity measures against the M&C Data 144
6.3 Comparison of CRMs against the M&C Data 145
6.4 Frequencies and precision and recall values (additive MI-based model) for the noun pairs in the M&C data 150
6.5 Mean correlation between distributional neighbour sets and WordNet derived neighbour sets. Figures in brackets are standard deviations 152

6.6 Proportion of pairs (n_1, n_2) where the recall of n_2's retrieval of n_1 is greater than its precision 159
6.7 Proportion of n_2s found in neighbour sets of size $k = 200$. We use the notation n_1 ISA n_2 to denote that the n_2s are hyperonyms of the n_1s. 161

Chapter 1

Introduction

This thesis is a study of measures of lexical distributional similarity and their applications. In particular, it investigates whether the characteristics that make a measure useful are the same in different applications.

In earlier research, lexical distributional similarity has been defined fairly loosely. Two words are said to be distributionally similar if they appear in similar contexts. This definition gives rise to a number of questions.

1. What is a "context"?
2. Are all contexts equally important?
3. Does it matter to what extent each word occurs in each context?

As is discussed in Chapter 2, following a number of earlier researchers (Hindle, 1990; Pereira, Tishby, & Lee, 1993; Lin, 1998a; Lee, 1999) we define "context" in terms of grammatical dependency relations. By taking a syntactic view of context, we see that distributional similarity can be thought of in terms of lexical substitutability (Church, Gale, Hanks, Hindle, & Moon, 1994): the distributional similarity of two words is the extent to which they can be inter-substituted without changing the plausibility of the sentence.

The concept of substitution, however, is inherently asymmetric. It is possible to measure the appropriateness of substituting word A for word B independent of measuring the appropriateness of substituting word B for word A. Similarity has been defined in terms of *inter-substitutability* but we ask whether there is something in the inherent asymmetry of substitution that can be exploited by an asymmetric measure of distributional similarity.

The idea of lexical substitutability also highlights the relationship between distributional similarity and semantic similarity. Semantic similarity can be thought of as the degree of synonymy which exists between two words, i.e., the extent to which they can be inter-substituted without changing the meaning of the sentence. Thus, it would appear that distributional similarity is a weaker requirement than semantic similarity since two sentences can be plausible without meaning the same thing, but cannot mean the same thing without both being plausible. In other words,

semantic similarity logically implies distributional similarity but distributional similarity does not logically imply semantic similarity.

However, the hypothesis that semantic similarity can be predicted from distributional similarity has led to work on automatic thesaurus generation (Hindle, 1990; Grefenstette, 1994b; Lin, 1998a; Curran & Moens, 2002a). The results are usually encouraging, but rarely conclusive due to the large number of available choices in distributional similarity measure selection and evaluation.

In order to explore the characteristics which make a useful measure of distributional similarity, we turn to its applications. These applications can be roughly divided into those which make use of the property of distributional similarity directly, e.g., language modelling, and those which use it as a predictor of semantic similarity, e.g., automatic thesaurus generation. It is our hypothesis that a single, unparameterised measure of distributional similarity will not be able to perform well in both of these application areas due to the inherent asymmetry in the concept of substitutability. Asked for a semantically related word to *dog*, we might say *animal* since *animal* can generally be used in place of *dog*; whereas we are less likely to say *dog* for *animal* since *dog* cannot generally be used in place of *animal*. This preference in the direction of the relationship between the two words is not necessarily maintained when one considers a distributional task, such as language modelling in the face of sparse data. If we want to learn what other contexts *animal* can occur in, then we might look at the co-occurrences of words such as *dog*, since we know that *dog* can generally be replaced by *animal*. If we want to learn what other contexts *dog* can occur in, we are less likely to look at the co-occurrences of *animal*, since we know that *animal* can occur in contexts in which *dog* cannot.

As a corollary of this, we would expect to find that corpus-derived distributionally similar words are more useful than manually-derived semantically similar words on the tasks that we classify as distributional. Conversely, manually-derived semantically similar words will perform better than automatically generated sets of similar words on semantic tasks, if the distributional similarity measure used in the automatic generation process is optimised for use in distributional tasks.

Our second hypothesis is that it is sufficient to know that the two words *can* occur in a particular context. Occurring in more of the same distinct contexts is far more important than occurring repeatedly in the same indistinct contexts. Once we have seen that *dog* can occur as the subject of *bark*, subsequent occurrences of *dog* as the subject of *bark* tell us little about *dog* or the words which are similar to it. As a result, we expect to observe that incorporating the difference in extent to which each word occurs in a shared context will not improve results. However, this is not to say that the frequency information is useless. Some verbs tell us more about their arguments than others, e.g. *bark* tells us more about its subject than *have*, and this can be estimated from the distributional information. We hypothesise that measures that give more weight to these more informative contexts will perform better than measures that give more weight to more frequently occurring measures.

Having outlined our major hypotheses, we now present a chapter-by-chapter summary of the remainder of this thesis.

Chapter 2: Background In this chapter, the background to our work is presented. We consider what it means for two words to be distributionally similar. We discuss different types of context,

orders of affinities between words and the relationship between distributional and semantic similarity. We then consider potential applications for distributional similarity techniques in Natural Language Processing (NLP). There are a large number of such applications which tend to exploit one (or both) of two principles. The first principle is that similar words tend to occur in similar contexts and so probability estimates can be formed over classes or clusters of similar words. The second principle is that naturally occurring texts tend to contain large numbers of semantically related words. Finally, we discuss possible approaches to evaluating similarity scores. We consider evaluation against a gold standard, evaluation against human plausibility judgements, evaluation against unseen data and evaluation using application-based tasks.

Chapter 3: Existing Measures of Distributional Similarity In this chapter, we discuss a broad range of existing distributional similarity measures that have been proposed or adopted for use in NLP. We present the theoretical basis for each and compare the measures in terms of the neighbours they produce from the same data. In particular, we examine their tendencies to select high or low frequency words as neighbours. Knowledge of these tendencies can allow us to compute neighbour sets efficiently. It may also give us insights as to the types of tasks for which a particular measure might be most useful.

Chapter 4: A General Framework for Distributional Similarity This chapter is the core of our thesis. In this chapter, we present a new general framework for distributional similarity. In this framework, similarity is viewed as a measure of the appropriateness of using one noun (or its distribution) in place of another and, as such, is inherently asymmetric. We can measure how appropriate it is to use word A instead of word B independently from how appropriate it is to use word B instead of word A. If one word's distribution is used in place of another, then the appropriateness of this can be viewed as how well the first distribution predicts or *retrieves* the co-occurrences of the second. In this way, the measurement of distributional similarity is cast as a *co-occurrence retrieval* (CR) problem, for which we can measure precision and recall by analogy with the way they are measured for document retrieval. We combine these precision and recall scores using parameters which can be tuned to a particular application.

We also define a number of different models for *co-occurrence retrieval*, which allow us to explore systematically a number of other issues in distributional similarity. We explore the relative importance of different contexts using type-based, token-based and MI-based models of co-occurrence retrieval. *Additive* and *difference-weighted* versions of each model are also provided, which allow us to investigate whether considering the difference in extent to which each word occurs in each context outperforms simpler models where contexts are simply shared or non-shared.

Having laid out the theoretical basis of our framework, we compare it with the existing measures of distributional similarity discussed in the preceding chapter. First, we examine the tendencies of different models and parameter settings to select high or low frequency words. Second, we attempt to characterise each of the existing measures in terms of framework parameter settings, i.e. their CR characteristics.

Chapter 5: Language Modelling In this chapter, we consider the application of distributional similarity measures in language modelling. Smoothing probability estimates for co-occurrences with those from nearest neighbours is an intuitively plausible way of overcoming the sparse data problem and, hence, providing more reliable probability estimates. We first demonstrate that this

is a feasible thing to do by deriving smoothed co-occurrence probability distributions using one distributional similarity measure, and considering their similarity with unseen empirical distributions. We then compare the co-occurrence retrieval models and some of the existing distributional similarity measures in terms of results on a pseudo-disambiguation experiment. Finally, we show that differences in performance on the pseudo-disambiguation experiment are reflected in differences in performance on the original evaluation against unseen empirical distributions.

Chapter 6: Predicting Semantic Similarity In this chapter, we consider the ability of distributional similarity to predict semantic similarity. We do this in four different ways. First, we correlate corpus-derived distributional similarity judgements with human synonymy judgements. Second, we compare neighbour sets produced by distributional similarity measures with neighbour sets derived from WordNet (Fellbaum, 1998). Third, we examine the relationship between the direction in our asymmetric distributional similarity framework and the direction in semantic similarity as defined by the relation of hyponymy in WordNet. Fourth, we consider performance by semantic and distributional similarity measures on a spelling correction task.

Chapter 7: Conclusions and Further Work In this chapter, we present our conclusions and a number of directions for further work.

Chapter 2

Background

In this chapter, we present the background to our work. We consider what it means for two words to be distributionally similar, potential applications for distributional similarity techniques and a number of possible approaches to evaluating similarity scores.

2.1 Lexical Distributional Similarity

One fairly loose definition of lexical distributional similarity is that two words are distributionally similar if they appear in similar contexts. This definition is in keeping with the Firthian tradition that "You shall know a word by the company it keeps" (Firth, 1957). A word is described in terms of the company it keeps, i.e. the contexts in which it occurs, and a measure of similarity is used to determine the similarity of two such descriptions.

Such a definition corresponds with what Grefenstette (1994a) terms a second order affinity between words. Two words have a first-order affinity if they appear in each other's context. A second-order affinity exists between two words if they tend to have the same first-order affinities. Grefenstette further defines a third-order affinity as existing between words which have similar second-order affinities. He illustrates the difference between first and second order affinities with the example of spelling variations such as *tumor* and *tumour*, which will have a very low first order affinity but a high second order affinity.

This result, however, is not surprising given the following definition of synonymy:

> Two words are absolute synonyms if they can be inter-substituted in all possible contexts without changing the meaning.

This definition dates back to von Leibniz (1704) who said, "eadem sunt quorum unum potest substitui alteri salva veritate."[1] More recently, this philosophy has been applied to language (Quine, 1960; Ullmann, 1962; Lyons, 1977; Church et al., 1994), where the two things under consideration are words and the "truth" referred to is the meaning of the sentence.

From this definition we can see that if two words cannot be inter-substituted in every possible context without changing the meaning, they cannot be synonyms. Distributional similarity tests

[1] Two things are identical if one can be substituted for the other without affecting the truth.

whether the inter-substitution of two words in all contexts leads to plausible sentences[2] but it does not test whether the truth conditions or the meaning of the sentence has been changed. Therefore, distributional equivalence (i.e. maximal distributional similarity) is a necessary condition for synonymy but it would not appear to be a sufficient condition for synonymy. Further, it is not difficult to see that distributional similarity cannot be a sufficient condition for synonymy since synonyms are not the only words which tend to be used in similar contexts. The two most distributionally similar words found by Lin (1998a) were the antonyms *rise* and *fall*. Another linguistic relation that tends to lead to high distributional similarity is the *hyponymy* relation. Hyponymy can be considered as the formalisation of the "ISA" relation. A *car* ISA *vehicle* and is therefore a hyponym of *vehicle*. Conversely, the super-ordinate term, *vehicle*, is referred to as a hyperonym or hypernym of *car*. A strict definition of hyponymy (Sparck-Jones, 1964) might attempt to exclude relationships based on factual knowledge about the world. Sparck-Jones argues that relationships based on world knowledge are encyclopaedic rather than linguistic in nature. However, as noted by Resnik (1993), WordNet (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990), which uses hyponymy as one of its main organising principles, considers *fruit* to be a hyponym of *groceries*. This type of hyponymy will also be seen amongst distributionally similar words since distributional similarity is based on usage in the world. First order affinities, however, seem to lie more within the domain of encyclopaedic knowledge. For example, the occurrence of the words *ball* and *net* in the same sentence in a sports report tells us about the world in which we live, not about the meanings of the words *ball* and *net*.

The difference between first and second order affinities is further discussed by Kilgarriff and Yallop (2000). They make a distinction between *looser* and *tighter* thesauruses. Looser thesauruses, such as Roget (Roget, 1911), include words with first order affinities whereas tighter thesauruses, such as WordNet (Fellbaum, 1998), include only words with second order affinities. This is because WordNet is organised using the linguistic principles of synonymy, hyponymy and antonymy.

With this background in place, we now consider the definition of distributional similarity in more detail. We said that two words are distributionally similar if they appear in similar contexts. This, however, raises a number of questions. First, what do we mean by "context"? Second, should multiple occurrences of a word in a distinct context be counted separately or do *n* occurrences of word A in context B tell us no more about the word than a single occurrence of word A in context B? Third, are all contexts equally important? We now consider each of these questions in turn.

Context can be modelled at a number of different levels. For example, we can consider the context of a word to be the document in which it occurs, an n-gram, a bag-of-words on either side or the words with which it has some grammatical dependency. The first three types of context are fairly similar and differ mainly in the amount of context considered. For example, a document can be considered as a very large bag-of-words. An example of a technique which uses this level of context is latent semantic indexing (LSI) (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). In LSI, each document in a database is described by a vector of the words, or terms, it contains and each word, or term, is described by a vector of the documents in which it occurs. From this, one can determine which words, or terms, are similar in respect of the documents in

[2]We use *plausible sentence* to refer to a sentence which might be observed in real language data.

which they occur, and hence retrieve documents which are relevant to a query even though they do not actually contain the words in the query.

The use of grammatical dependency relations in determining context, however, requires annotated text, and leads to significantly different results to the use of any of the first three types of context. Kilgarriff and Yallop (2000) note that a second order automatic thesaurus which uses grammatical relation data to determine context (Lin, 1998a) tends to generate sets of words which occur in the same semantic class. Such a thesaurus is a "tighter" thesaurus which explains why Lin found his thesaurus entries more akin to WordNet than Roget. Conversely, a technique such as LSI, which uses document-level context, provides a classification which is more akin to Roget.

The reason why using *bag-of-words* level context leads to looser thesaurus entries than using grammatical dependency relations as context can be seen by considering an example:

> Emma Woodhouse, handsome, clever, and rich, with a comfortable home and happy disposition, seemed to unite some of the best blessings of existence; and had lived nearly twenty-one years in the world with very little to distress or vex her. (Austen, 1816)

Considering evidence from the start of this sentence alone, we can see that the adjectives *handsome*, *clever* and *rich* all occur together and are all used to modify *Emma Woodhouse*. If we model context as the three words on either side of a word, we will find that *handsome* and *clever* share some of the same context words {*Emma, Woodhouse, and, rich*} but we will also find that *Woodhouse* and *handsome* share some of the same context words {*Emma, clever, and*}. We might expect that future occurrences of *Woodhouse* will not appear in the same contexts as *handsome* and therefore their similarity will diminish as more evidence is seen (not, however, if Emma is referred to as "the handsome Miss Woodhouse"). But, in any case, the fact remains, that some amount of second order affinity will be established between all words with first order affinities if this bag-of-words model of context is used. One way of overcoming this is by considering bigrams or a single word on either side as context. However, such limited context is not always very informative. In the above example, the contexts of *rich* would be *and* and *with*, which do not seem to tell us much about the word *rich*. Alternatively, we can consider grammatical dependency relations. Knowing that *handsome*, *clever* and *rich* all modify *Emma Woodhouse* will lead to a certain degree of similarity between those adjectives but will not lead to *Woodhouse*, *Emma* or the phrase *Emma Woodhouse* being similar to any of them. Instead, these words will exhibit similarity with other things that are described has *handsome*, *clever* or *rich* later on in the novel.

The use of grammatical relation data to model context is not new. The claim of the distributional hypothesis (Harris, 1968) is that:

> The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities.

This distributional hypothesis has been the subject of much debate. Does meaning dictate behaviour or does behaviour determine meaning? Keil (1979, 1983) argues that children learn the hierarchical structure of nominal concepts by observing what can and cannot be predicated at each level. Levin (1993) takes the opposite approach and shows that verb participation in diathesis alternations can be predicted from their semantic class. Rubenstein and Goodenough

(1965) and Miller and Charles (1991) investigate the link between behaviour and meaning by correlating distributional similarity scores with human synonymy judgements. However, it was probably Hindle (1990) who was first able to demonstrate the plausibility of the distributional hypothesis. Previous similar research (Hirschman, Grishman, & Sager, 1975) had been limited by the amount of grammatically annotated text that could be processed. Using a robust syntactic analyzer (Hindle, 1983) to extract grammatical relation data from a six million word sample of Associated Press news stories and a measure of similarity based on mutual information, Hindle (1990) was able to show the plausibility of deriving semantic relatedness from the distribution of syntactic forms.

A large number of subsequent researchers take a similar approach and use grammatical relation data as their model of context. For example, Pereira et al. (1993) propose using noun clusters based on distributional similarity for smoothing. There are a large number of subsequent publications (Lee, 1997; Dagan, Lee, & Pereira, 1999; Lee, 1999; Lee & Pereira, 1999; Lee, 2001) that extend this work. Lee and Pereira (1999), for example, investigates different clustering techniques. Lin (1998a) automatically generates a thesaurus based on grammatical relation data. Caraballo (1999) takes a similar approach to generate initial clusters of nouns and then attempts to determine hyponymy relations within the clusters from other corpus data. There is also a large body of work which attempts to derive Levin verb classes (Levin, 1993) from distributional data. Dorr and Jones (1996) cluster verbs semantically based on the diathesis alternation behaviour given by Levin. Lapata and Brew (1999) use syntactic frame information to disambiguate verbs which occur in more than one of Levin's classes. Stevenson and Merlo (1999, 2000) induce a semantic classification of verbs based on diathesis alternations. Schulte im Walde (2000) clusters German verbs semantically according to subcategorization and selectional preference information derived from grammatical relation data.

However, thirteen years later, the debate on which level of context to use still rumbles on. Over the years, different levels of context have been shown to be useful in different applications. For example, Clark (2000) uses just the preceding word as context in order to automatically induce syntactic categories. This captures certain intuitions about word order e.g. determiners often precede nouns. Further, such an application could not be based on grammatical relation data since producing grammatical relation data is based on the syntactic categories that Clark is attempting to induce. Vector space models based on bag-of-words level context have been shown to be useful in information retrieval tasks (Salton, Wang, & Yang, 1975) since we would expect the extra associations captured by this level of context to be useful in topic identification. Padó and Lapata (2003) use the concept of a dependency path to show that a vector space model based on grammatical relation data can be used to differentiate between first order affinity or association relations (conceptual association and phrasal association) and second order affinity or taxonomy relations (synonymy, hyponymy, category co-ordination and antonymy), thus providing more evidence of the differences between these two types of relation.

Although we may not have heard the end of the debate on the context representation to be used, the distinctions between the *bag-of-words* level and the *grammatical relations* level are fairly well understood. Accordingly, this work will look at what can be learnt from grammatical relation data. We concern ourselves with what can be learnt about syntactically substitutable words. From this

starting point, we focus on the second two questions given above, which can be summarised as, "Given descriptions of words based on their co-occurrences with other words in certain grammatical relations, how do we calculate the similarity of two words?"

A multitude of measures of similarity have been adopted or proposed for this purpose and we shall look at these in detail in Chapter 3. There has been previous work on evaluating a number of these measures. Lee (1999) uses a pseudo-disambiguation task to evaluate eight measures. Lin (1998a) considers six measures in his comparison with neighbour sets derived from WordNet and from Roget. Most recently, Curran and Moens (2002a) compare 9 different measures and 9 different weight functions in an evaluation against a combination of three thesauruses (the Macquarie (Bernard, 1990), Moby (Ward, 1996) and Roget (Roget, 1911)).

However, most of the work has been interested in finding the "best" similarity measure and little work has been done on classifying the measures. One classification that has been made (Lee, 1999) considers the influence of how the information from the intersection of the supports of the two distributions is incorporated. Lee's conclusion is that measures which focus on this intersection, i.e. the co-occurrences which both words share, do better than measures which take into account information from co-occurrences outside the intersection.

In this work, we attempt to answer three further questions about the calculation of distributional similarity. First, is a shared context worth the same regardless of the difference in the extent to which each word appears in that context? In other words, should multiple co-occurrences of the same context be treated as distinct? One argument is that a single co-occurrence of two words is sufficient evidence to say that those two words can occur together. A single co-occurrence of the words *animal* and *bark* may be considered sufficient evidence[3] to say that *animal* can replace the word *dog* in the context of *bark*, regardless of the number of times *dog* and *bark* have co-occurred. However, a counter-argument is that substitution by a word that has a similar probability in that context may make a better substitution.

Second, should all contexts be treated equally or are some contexts more important than others? It may be that more frequently occurring contexts should be considered more important. For example, a single co-occurrence of *animal* and *bark* may not be sufficient to make us worry whether our substitute for *animal* can occur in the context of *bark*. Alternatively, it may be that contexts that occur with a smaller number of concepts are more important. For example, *animal* may occur very frequently in the context of *have* but this is true for the majority of nouns. Verbs such as *have* are selectionally weak (Resnik, 1993) since they can be applied to many different words or concepts. Such verbs do not tell us very much about the behaviour or meaning of *animal* and therefore we might consider giving them less weight in our similarity calculation.

Both of these questions are concerned with how the frequency information of different contexts are utilised by the measures. In Chapter 3 we consider how existing similarity measures utilise such frequency information. In Chapter 4, we present a general framework in which we can explore these issues systematically.

The last, and possibly most important, issue that we address is the one that has probably received the least attention in the literature. The vast majority of existing similarity measures assume that similarity is a symmetric property. However, it is our hypothesis that similarity should

[3] In the presence of noise, one may want to raise the required number of co-occurrences to a threshold value greater than 1.

be viewed as asymmetric, that is word A can be similar to word B without word B being similar to word A. Viewing similarity as a measure of substitutability, it may be appropriate to substitute word A for word B when it is not appropriate to substitute word B for word A. For example, *animal* may make a good substitute for *dog* whereas *dog* may not be a good substitute for *animal*:

The dog howled all night.

The animal howled all night.

The animal mewed pitifully.

? The dog mewed pitifully.

Asymmetry in similarity is raised by Lee (1999) in her justification of her α-skew divergence measure, which is based on the asymmetric Kullback-Leibler or relative entropy measure. However, Lee's work does not explore fully how asymmetry might be exploited. We will consider symmetry in existing measures in Chapter 3 and present our own inherently asymmetric framework in Chapter 4.

We have, so far, considered a number of intuitions about distributional similarity and its calculation, which will be investigated throughout the rest of this thesis. Now, however, we turn our attention to the question of why we want to calculate distributional similarity. The answer to this question not only motivates work on distributional similarity but can also provide insights as to its measurement and evaluation.

2.2 Applications

There are many potential applications of distributional similarity measures. In this section, we survey and classify fifteen potential applications, ranging from prepositional phrase attachment ambiguity resolution to Information Retrieval. We shall first discuss a classification of applications which roughly divides them into distributional (or syntactic) applications and semantic applications. We will then consider each application in turn, discussing how distributional similarity information might be utilised. Finally, we shall briefly discuss clustering techniques, a topic which is relevant to all of the applications.

2.2.1 Classification of Applications

The fifteen potential applications of distributional similarity measures can be roughly divided into two categories: those which use the property of distributional similarity directly and those which use the property of distributional similarity as a predictor of semantic similarity. In other words, some fall in the syntactic domain and others in the semantic domain. Our classification of potential applications is shown in Table 2.1. The first application in each category is a general application which characterises that category. The other applications are more specific and can make use of distributional similarity information within the paradigm of the characterising application.

Some of the applications that we classify as distributional have previously been tackled using semantic class information, and therefore may be thought to be semantic applications. For example, Resnik (1993, 1998) generalises lexical preferences in the prepositional phrase attachment problem over semantic classes in WordNet. Resnik gives a number of reasons for not using

Distributional Applications	Semantic Applications
Language modelling (Section 2.2.2)	Automatic thesaurus generation (Section 2.2.9)
PP attachment ambiguity resolution (Section 2.2.3)	Information Retrieval (Section 2.2.10)
Conjunction scope identification (Section 2.2.4)	Anaphor resolution (Section 2.2.11)
Word sense disambiguation (Section 2.2.5)	Collocation extraction (Section 2.2.12)
Word sense separation (Section 2.2.6)	Text simplification (Section 2.2.13)
Metonymy resolution (Section 2.2.7)	Compound noun interpretation (Section 2.2.14)
Learning countability preferences (Section 2.2.8)	Topic identification (Section 2.2.15)
(Spelling correction (Section 2.2.16))	Spelling correction (Section 2.2.16)

Table 2.1: Classification of potential applications of distributional similarity measures according to whether they primarily make use of distributional information or semantic information

distributional similarity classes in this task. He argues that distributional similarity classes are 1) expensive to compute; 2) do not have symbolic labels; 3) have no clear semantics; 4) focus on word forms rather than senses; and 5) are over-fitted to a particular corpus. However, for the tasks that we classify as *distributional*, we can counter each of these arguments as follows:

1. Recent increases in hardware capability mean that we now have the computational power to be able to compute distributional similarity classes based on large corpora such as the British National Corpus (BNC).

2. It is not necessary for a group or class of objects to have a name for us to know that they are similar and to exploit that information. For example, we can make use of the knowledge that *cats* are similar to *dogs*, without knowing that both are part of a larger class known as *animals* or *mammals* or *pets*.

3. In the applications we classify as distributional, the use of semantic class information is used to predict distributional information. Accordingly, it is not obvious why a class should need a clear semantics in order to be able to do this. If all of the objects in the class behave in the same way distributionally, e.g., they can be modified by the same prepositions, then it would not seem to matter whether they mean similar things.

4. Conflation of word senses in corpus data may be a problem for distributional techniques. However, if we have sense-tagged corpus data, we can use distributional techniques on word senses rather than word forms. If, on the other hand, we do not have sense-tagged corpus data, we cannot exploit it to solve structural ambiguities, such as prepositional phrase attachments. Accordingly, probabilities based on word forms rather than word senses may in fact be more useful.

5. Domain or dialect specific information gained from deriving distributional classes from a particular corpus might be beneficial if the corpus is representative of the sentences to be disambiguated.

Consequently, in tasks where semantic classes have previously been used to predict distributional information, there is no reason why we should not do as well, if not better, using distributional similarity classes. The tasks that we classify as *semantic* are ones which rely on the distributional hypothesis: that semantics can be predicted from syntax. In these cases, we might expect a technique that uses semantic classes to do better than one that uses distributional classes. However, if the distributional hypothesis is true, then we might be able to gain an advantage by using distributional techniques on domain or dialect specific corpora. We will now discuss each potential application in turn, starting with the distributional applications.

2.2.2 Language Modelling

Language models are essential for statistical natural language processing techniques. For example, in speech recognition, the likelihood that the current word is X depends both on the signal and the prior probability of the word. The prior probability of a word, or our expectation of hearing that word, depends on our experience of language. We may use global knowledge, that some words are more likely than others, or local contextual knowledge, that in the current context, one word is more likely than another. In statistical parsing, the probabilities of alternative parses depend on prior probabilities of constituent constructions i.e. our expectation of seeing that construction. Again, this expectation depends on global and contextual knowledge about the use of constructions. Similar techniques are applied to address ambiguity in nearly every area of natural language processing. In all cases, a language model is required to define the prior probabilities of events and co-occurrences of events. For example, an n-gram model, often used in speech recognition, defines the conditional probability of a word given the previous $n-1$ words. The problem is how to acquire this language model. Language is infinite and therefore it is not possible to write down every possible combination of words, let alone calculate the probability of them being uttered or written. This problem is exacerbated by the Zipfian nature of text (Zipf, 1949). In any particular corpus, there are a small number of events that occur very frequently and a large number of events, and even larger number of co-occurrences of events, that are never seen at all. In other words, maximum likelihood estimates made on the basis of the corpus data will over-estimate the probabilities of seen events at the expense of unseen events. A statistical technique that uses a language model which assigns a zero probability to these unseen events, will rule the correct parse or interpretation of the utterance impossible when they eventually do occur. Further, although an individual unseen event may have a very small probability of occurring in the future, the cumulative effect of all the unseen events is much greater. For example, Brown, DellaPietra, deSouza, Lai, and Mercer (1992) report that one can expect 14.7% of the word triples in any new English text to be unseen in a training corpus of 366 million English words. Consequently, smoothing techniques (Katz, 1987; Laplace, 1995; Gale & Sampson, 1996; Chen & Goodman, 1998; Friedman & Singer, 1999) to overcome this sparse data problem are the focus of much attention in language modelling.

Probably the simplest approach to smoothing is naïve Bayesian estimation, known more commonly as Add1 smoothing. Add1 smoothing overcomes the zero probability problem by assuming that every configuration of events (e.g. word triple) has been seen once before the training data is considered. This approach, however, tends to assign too much probability mass to unseen events and so, Add1 often becomes AddX where X is less than 1. Further, AddX can be combined with techniques like Simple Good-Turing Estimation (Gale & Sampson, 1996), which estimates the amount of probability mass that should be assigned to unseen events based on the number of hapax legomena (events that occur only once) in the data. However, although such a technique allows for the possibility of unseen events occurring, it does not allow us to disambiguate between them since they are all assigned the same probability. Back-off smoothing (Katz, 1987) overcomes this by making an independence assumption: the probability of an unseen co-occurrence is proportional to the probability of one or both of the individual events. However, in some ways, this is counter-intuitive. If two very common words have never been seen to co-occur, this might actually be good evidence that they cannot co-occur together. It is the distributions of the rarer

words, which we have seen less often, in which we have less confidence and therefore it these distributions that should be subject to greater change.

Similarity-based smoothing (Hindle, 1990; Brown et al., 1992; Dagan, Marcus, & Markovitch, 1993; Pereira et al., 1993; Dagan et al., 1999) offers an intuitively appealing alternative. The probabilities of unseen co-occurrences are estimated from the probabilities of seen co-occurrences of distributionally similar words. The only assumption this approach makes is that words which have been seen to share many co-occurrences are likely to actually share more co-occurrences.

2.2.3 Prepositional Phrase Attachment Ambiguity Resolution

The prepositional phrase attachment problem is discussed in detail by Resnik (1993). The task is to decide whether a prepositional phrase attaches to the verb or the noun. Early work (Hindle & Rooth, 1991, 1993) on lexical preference relationships for this task concentrated on the probability of the preposition given the verb and the probability of the preposition given the noun. For example:

(1) Mary liked their house in the country.

(2) Mary visited their house with John.

In the first sentence, the prepositional phrase clearly attaches to the noun phrase “their house”. We can make this judgement based on our intuition, or language model, which tells us that a house is more likely to be in somewhere than something is to be liked in somewhere. In the second sentence, however, the prepositional phrase clearly attaches to the verb: the prepositional phrase tells us who accompanied Mary on her visit. Accordingly, we might expect to be able to determine this from considering whether the probability of the preposition given the verb is greater or less than the probability of the preposition given the direct object of the verb. However, as discussed by Hindle and Rooth (1993) and Resnik (1993), the prepositional object also plays a part. Imagine we were to say instead:

(3) Mary visited their house with a gold roof.

In this case, “they” have more than one house, one of which has a gold roof and this was the one that Mary visited. However, if we try to build a statistical model on the basis of four lexical events, we are faced with extremely sparse data. Even in a very large corpus, we might never have seen a mention of a “house with a roof” (since it might normally be assumed that a house has a roof without explicit mention of it). And even if we have seen, say, a single reference to a “house with a thatched roof”, how do we know that this is really more likely than “visiting with a roof”, when smoothing tells us to assume nothing is impossible?

Recent classifiers for the PP-attachment problem have made use of techniques like Maximum Entropy (Ratnaparkhi, Reynar, & Roukos, 1994), decision lists (Collins & Brooks, 1995) and boosting (Abney, Schapire, & Singer, 1999). These techniques overcome the data sparseness problem by making decisions based on more frequently occurring features (e.g. “the preposition is with”) in the absence of more finely tuned features.

An alternative approach (Resnik, 1993; Clark & Weir, 2000) is to induce probability distributions over semantic classes rather than lexical items. For example, a house is a type of building,

seeing the co-occurrence of any type of building and "with a roof" e.g. "a cottage with a thatched roof", may strengthen our belief that the prepositional phrase attaches to the noun in this instance. Similarly, we can consider the semantic class of the verb and the prepositional object. For example, "John" is a *human* and visit is a *motion* verb. By accumulating co-occurrences over these semantic classes, more reliable probability estimates can be formed.

However, these techniques are using semantic information to predict what is essentially distributional information. Rather than predicting what words co-occur together on the basis of clusters of semantically similar words, why not predict what words co-occur together on the basis of cluster of words which tend to co-occur with the same words? Stetina and Nagao (1997) report state-of-the-art performance on the PP-attachment ambiguity resolution problem (88.2%) using a semantic dictionary but other techniques which use semantic information have not been so successful. Both Clark and Weir (2000) and Li and Abe (1998) use WordNet based techniques for smoothing and report performance levels of around 80%. Li (2002), however, shows that a distributional technique outperforms a WordNet-based technique in terms of accuracy, but not in terms of coverage. Li hypothesises that the reason for the increase in accuracy is due to the domain-dependent nature of the corpus (Wall Street Journal) and that the reason for the lack of coverage is due to the size of the corpus (126,084 sentences). By combining automatically constructed word classes with WordNet, they achieve the same level of performance as Stetina and Nagao (1997) (88.2%). Pantel and Lin (2000a) report performance of 84.3% using an unsupervised approach to prepositional phrase attachment based on distributional similarity techniques. This significantly outperforms other unsupervised techniques and is drawing close to the state-of-the-art supervised techniques.

2.2.4 Conjunction Scope Identification

Another form of structural ambiguity that can be tackled using similarity methods is that of conjunction scope. Resnik (1993) investigates noun co-ordinations of the form *noun1 and noun2 noun3*, such as:

(1) John is a ((business and marketing) major).

(2) John is an (athlete and (economics major)).

Kurohashi and Nagao (1992) note that form and meaning play a role in determining conjoinability. For example, the plurality of the nouns being conjoined should match. In the above examples, however, plurality of all three nouns is the same and therefore cannot be used. In these cases, it is the semantic similarity of the conjoined heads that determines the correct structure: *business*, *marketing* and *economics* are academic subjects whereas *athlete* and *major* are words for people. Using semantic similarity instead of form similarity, Resnik (1993) finds that coverage increases from 40.4% to 69.7% whilst accuracy drops from 80.6% to 77.4% for a set of 100 noun phrases extracted from the Wall Street Journal. By backing off from form similarity to semantic similarity, Resnik obtains 85.4% coverage and 81.6% accuracy.

However, we note that this is another case where distributional similarity might be used directly rather than as a substitute for semantic similarity. The preferred structure can be determined by considering the probability with which the first noun can modify the third noun. In the example, we can consider the probability of seeing "business major" and "athlete major". This recasts

the problem as a language modelling problem, where distributionally similar words can be used to overcome the sparse data problem. This approach can also be applied to co-ordinations of the form *adj noun1 and noun2* where we can consider the likelihood of the adjective modifying the second noun:

(3) He only eats juicy apples and ice-cream.

(4) He only eats juicy apples and oranges.

2.2.5 Word Sense Disambiguation

Word sense disambiguation (WSD) is the determination of the meaning, or sense, of a word in context. For example, the noun "plane" has at least four senses which we will characterise as *aeroplane*, *tool*, *geometric surface* and *tree*:

aeroplane The plane circled over head.

tool John knew he'd find a plane in the toolshed.

surface The distance between two points on a plane can be easily calculated.

tree The plane was blown down in the hurricane.

Being able to tell which sense is intended, is essential in the understanding or translation of these sentences. Many word sense disambiguation algorithms (Lin, 1997; Leacock & Chodorow, 1998; Schütze, 1998; Karov & Edelman, 1998; Brill, 2000; Kilgarriff & Tugwell, 2001a) rely, either implicitly or explicitly, on Yarowsky's principles (Yarowsky, 1993, 1995) of "one-sense-per-collocation" and "one-sense-per-discourse". For example, every time the word "plane" appears as the subject of the verb "circle", we might expect that it is the *aeroplane* sense intended. Further, in Yarowsky's WSD algorithm (Yarowsky, 1995), having labelled this occurrence as being the *aeroplane* sense, other constraining contexts such as "over head" might be learnt from the same occurrence. Occurrences of ambiguous words in less constraining local contexts can be disambiguated on the basis that it is extremely likely that the same sense is intended as in the rest of the discourse, i.e., previous (or possibly subsequent) occurrences of the same word that can be disambiguated based on local context. If all of these approaches fail, global or domain information can be used to decide which sense is most likely.

This algorithm, however, will not work without some initial input or *seeds*. These seeds are the original, known constraining contexts which can be labelled. Yarowsky uses dictionary definitions to supply seeds to his algorithm. Alternatively, Kilgarriff and Tugwell (2001a) extract salient contexts from the corpus using mutual information. These contexts are displayed to a user who identifies the sense of some or all of them. These are then used as the seeds for Yarowsky's WSD algorithm.

An alternative approach assumes that a particular sense of a word will occur in similar contexts to a word that defines that sense. For example, Karov and Edelman (1998) sense tag a training corpus by considering the contexts shared by the target word and its sense words (words which appear in the dictionary definition of the target word). For example, contexts of "suit" shared by "clothes", e.g. "wear", may be tagged as the clothes sense of suit, whereas contexts of "suit" shared by "court" may be tagged as the court sense of suit.

This idea can also be applied in the other direction. Lin (1997) finds words which occur in the identical local context to the ambiguous target word, and selects the sense of the ambiguous word which maximises semantic similarity in WordNet between that word and the contextually similar words. Pantel and Lin (2000b) apply a similar approach to word-for-word glossing (which can be viewed as a word sense disambiguation task where the sense tags are foreign words). They find the contextually similar words and then maximise similarity between the set of contextually similar words and translations in a bilingual dictionary.

Finally, Schütze (1998) dispenses with the need for an external source of word sense information by tackling the related problem of word sense discrimination. Schutze represents a word occurrence (or token) by a vector containing all of the words occurring in its context (e.g. sentence). He then applies the hypothesis that similar context vectors indicate that occurrences belong to the same sense using an agglomerative clustering algorithm.

One way to use distributional similarity information (as opposed to contextual similarity information which applies to tokens rather than types) in word sense disambiguation is in generalizing context probabilities over classes. Due to the sparse data problem, we might never have previously seen any sense of plane occur in the current local context, the subject of "circle". However, we might have seen the word "bird" occur as the subject of "circle". This can be used to determine the sense of "plane" if we know that "bird" and the *aeroplane* sense of "plane" tend to occur in similar contexts (e.g. as the subject of "fly" and "land"). This similarity can be inferred indirectly from semantic or taxonomic information or it can be determined directly using distributional similarity techniques on sense-disambiguated data. In this way, the word sense disambiguation problem is recast as a language modelling problem, where we are modelling the probabilities of concepts rather than words. In the next section we consider how the burdensome requirement of sense-disambiguated training data might be eliminated.

2.2.6 Word Sense Separation

In the previous section, we discussed word sense disambiguation as an application for similarity-based methods. However, the technique described required a sense-tagged training corpus, which limits the appeal of such a technique. We will now consider the possibility of being able to automatically identify similarity between word senses as opposed to word forms.

Lin (1998a) proposes a technique based on the idea that neighbours of the same sense will tend to be closely related to each other. Lin considers whether the next neighbour of the target word is more similar to the target word or to a previous neighbour of the target word. If it is more similar to a previous neighbour, it is assigned to the same sense as that neighbour; otherwise, it is assigned to a new sense. The technique, although simple, gives promising results. Once neighbours of a word have been organised into sense clusters, this information can be used instead of the source of semantic information in a WSD algorithm. For example, we can adapt the idea from Karov and Edelman (1998), discussed above, and use the neighbours of individual senses to assign particular co-occurrences to those senses.

2.2.7 Metonymy Resolution

Metonymy is a figure of speech, in which one expression is used to refer to the standard referent of a related one (Lakoff & Johnson, 1980). For example, the name of a place can often be used to refer to a sports team of that place:

England won the World Cup.

Since related words tend to have the same metonymic readings, metonymy can be viewed as a highly predictable form of polysemy (Kilgarriff, 1992). Accordingly, metonymy resolution can be cast as a word sense disambiguation problem where the standard referent of the related expression is viewed as a metonymic sense of the original expression. Lapata and Lascarides (2003) note that the intended paraphrase of the word (e.g., "England's rugby team") is usually so predictable that it may not be attested in corpus data due to Grice's maxim of brevity (Grice, 1957). They learn the most probable interpretation of a metonym by considering the probabilities of possible interpretations in constituent parts of the context. For example, they learn that "to enjoy the book" usually means "to enjoy reading the book" by considering the probabilities of "to enjoy reading" and "to read the book".

Markert and Nissim (2002) and Nissim and Markert (2003) investigate exploiting the regularity in metonymy using a supervised machine learning technique. From tagged data, a system can learn that "won" is a context in which "England" refers to a national sports team of England. Nissim and Markert (2003) use word similarity in two ways. The first generalises "England" to other nouns in the same semantic class. The second generalises the context "won" to other distributionally similar contexts (e.g., "lost"). In the second case Nissim and Markert (2003) use an automatically constructed thesaurus (Lin, 1998a)). However, as will be seen later, it might be feasible to identify the semantic class of the noun based on distributional techniques. Further, other distributionally similar nouns or phrases e.g. "the rugby team", might be used to identify the metonymic sense of the original word.

2.2.8 Learning Countability Preferences

Nouns in English can typically be classified as countable or uncountable[4]. For example, *dog* is countable, *furniture* is uncountable and *sugar* has countable and uncountable senses. Knowledge of countability preferences is important in language analysis, generation, translation and dictionary construction (Baldwin & Bond, 2003a). For example, Japanese does not mark countability, which means that the generation component of a Japanese-English translation system has to decide the countability of the noun in English (Bond, 2001).

One of the features (Baldwin & Bond, 2003a, 2003b) which makes it possible to classify nouns as countable or uncountable is their distribution with respect to determiners and numerical quantifiers. For example, we say "a dog" and "two dogs" but not "much dog" or "some dog". In contrast, we refer to "some furniture" but not "a furniture" or "two furnitures". Accordingly, if nouns are clustered based on their distributional similarity with respect to determiner and numerical quantifier contexts, we would expect to find distinct clusters corresponding to countability classes.

[4]Uncountable nouns are also known as mass nouns.

2.2.9 Automatic Thesaurus Generation

Let us consider the definition of synonymy introduced in Section 2.1:

> Two words are absolute synonyms if they can be inter-substituted in all possible contexts without changing the meaning.

Many philosophers and linguists (Quine, 1951; Goodman, 1952; Cruse, 1986; Clark, 1992) have argued against the existence of absolute synonyms and it is generally accepted that if an absolute synonym did exist, it would either fall into disuse or take on a new nuance of meaning (Clark, 1992).

However, even if no two words can ever completely fulfil the requirements of absolute synonymy, we can measure the extent to which the requirements are fulfilled, which gives rise to the concepts of near-synonymy and semantic relatedness. Accordingly, the semantic relatedness between two words can be viewed as the extent to which those words can be inter-substituted without changing the meaning.

One consequence of this definition of semantic relatedness is that we would expect semantic related words to behave in similar ways. For example, Resnik (1993) shows that the selectional preferences of verbs can be established over classes of semantically related nouns. Resnik's hypothesis is that the distribution of nouns and verbs can be predicted from their semantics (e.g. the things which can be eaten will occur as the object of the word eat).

The hypothesis in automatic thesaurus generation, however, is in the opposite direction: the semantics of nouns and verbs can be predicted from their distributions (e.g. the things which occur as the object of the word eat are things which can be eaten). In other words, if two words can be inter-substituted, we would expect to find them occurring in similar contexts, i.e. we would also expect them to be distributionally similar. This is the basis for a large body of work on automatic thesaurus generation (Hindle, 1990; Grefenstette, 1994a; Lin, 1998a; Curran & Moens, 2002a). There are inherent problems in evaluating automatic thesaurus extraction techniques, which will be discussed in Section 2.3, but recent results have been promising. Lin (1998a) shows that automatically generated thesaurus entries derived using his distributional similarity measure are more similar to WordNet derived thesaurus entries than Roget derived entries are. More recently, Curran and Moens (2002a) show that a combination of the Dice Coefficient and a T-Test weight function (Manning & Schütze, 1999) produce thesaurus entries which are more similar to a semantic gold standard (constructed by the authors from the Macquarie (Bernard, 1990), Moby (Ward, 1996) and Roget (Roget, 1911)) than when a number of other distributional similarity measures and weight functions are used (including that of Lin (1998a)). However, the lack of a consistent evaluation methodology between these pieces of research, makes it difficult to draw conclusions. Curran and Moens (2002a) evaluate over a much smaller number of words than Lin (1998a) (70 nouns compared to 4294 nouns) and a different semantic gold standard is used. Whether we want thesaurus entries that are more similar to WordNet derived entries or more similar to Roget-style entries may depend on the application. Given our earlier discussion of the differences between WordNet and Roget in terms of first and second order affinities, it would be surprising if one distributional similarity measure was best at producing entries most similar to both.

We will return to the questions of evaluation in Section 2.3. For now we will assume that, given an ideal distributional similarity measure, we can determine sets of semantically similar

words. If this hypothesis is correct, then we can quickly and easily build or extend thesauruses for a particular language, dialect, genre or domain. Domain-specific knowledge is being increasingly recognised as important in natural language understanding tasks. For example, word sense ambiguity can be reduced by considering the domain (Maynard & Ananiadou, 1998). In the domain of computing, the word or term *worm* is more likely to refer to a "malicious computer program" than an "earthworm". This domain knowledge will be reflected in a thesaurus automatically generated from a computing specific corpus, which will show increased similarity between *worm* and *virus* and reduced similarity between *worm* and *caterpillar*.

However, semantic relatedness as might be determined by distributional similarity methods encompasses several different semantic relations including synonomy, antonymy and hyponymy (Church et al., 1994). In order to be able to rival manually constructed thesauruses, automatically generated ones need to be able to distinguish between these different relations. There is also work (Copestake, 1990; Vossen & Copestake, 1993; Weeds, 2000) on genus-term or hypernym extraction from dictionary definitions. The structure of dictionary definitions is often such that the hyponymy relation is implicit or explicit between the word being defined and the head word of the defining noun phrase. For example, "adder: a poisonous snake" and "bitter: a type of dark, brown beer". Hence, distributional similarity could be used to enhance the genus-term extraction process in a system based on a machine-readable dictionary. However, such an approach could also be applied to corpus data. A system could search a large general corpus for corresponding phrasal patterns (e.g. "an adder is a poisonous snake") between words known to be distributionally similar. This is the approach taken by Hearst (1992) who searches for phrasal patterns that are well-known to express specific lexical relationships. For example, from a phrase such as "cats, dogs and other animals", we can identify "animal" as a likely hypernym for both "cat" and "dog". This idea was extended by Caraballo (1999) who first clustered nouns according to distributional similarity and then attempted to distinguish the hypernyms within these clusters. Lin, Zhao, Qin, and Zhou (2003) attempt to distinguish synonyms and antonyms using a similar approach. They propose that the phrasal patterns "from X to Y" and "either X or Y" are likely indicators of antonymy. They also hypothesise that antonyms are unlikely to have many translation words in common in a bilingual dictionary.

2.2.10 Information Retrieval

The primary function of a search engine is to retrieve documents that are related to the query. A typical approach, as discussed by Strzalkowski and Carballo (1993), is to index documents by key terms and simply return documents which have key terms exactly matching the query terms. However, this penalises documents (or queries) that use synonymous or subsuming terms rather than exactly matching terms. Query expansion (e.g. Xu and Croft (1996)) is a technique that adds other conceptually similar terms or phrases to the user's query; and, hence, requires a thesaurus or other source of semantic information.

2.2.11 Anaphor Resolution

Anaphor resolution is related to metonymy resolution in that a word or expression is used to refer to the referent of another related word or expression. However, in this case, the related word or

expression occurs as an antecedent in the text. For example, a synonym, hyponym or hyperonym may be used as a bridging anaphor:

(1) I bought a dog from a man on the street. My new pet felt at home immediately.

In this example, we know that a dog can be a pet and therefore infer that the pet referred to in the second sentence is the dog in the first sentence. The relationship between dog and pet could be derived from a semantic resource such as WordNet or using distributional techniques.

Techniques based on semantic similarity have also been applied to associative anaphora (Meyer & Dale, 2002). In an associative anaphor, the referent is not explicitly stated but implied, as in:

(2) The bus came round the corner. The driver had a nasty glint in her eye.

We can resolve this anaphor because we know that buses have drivers. This relationship between bus and driver can be learnt from expressions in corpora such as "the driver of the bus". However, in order to overcome the sparse data problem, Meyer and Dale (2002) generalise from bus to vehicle using the WordNet noun hyponymy hierarchy. We envisage a similar technique where distributional similarity methods are used to determine a cluster over which to generalise or where nearest neighbours are consulted.

2.2.12 Collocation Extraction

There are various definitions for the term collocation. Some researchers (Kilgarriff, 1992; Lin, 1998b) define a collocation as a habitual or recurrent word combination. Collocations, in this sense, can be extracted by considering relative word combination and unigram frequencies (Berry-Rogghe, 1973; Shimohata, Sugio, & Nagata, 1997; Blaheta & Johnson, 2001). More specifically, many researchers use mutual information as a measure of word association for this purpose (Church & Hanks, 1990; Kilgarriff, 1996; Lin, 1998b; Kilgarriff & Tugwell, 2001b). However, many researchers assign the term collocation a more specific sense, that is a lexicalised recurrent word combination or multiword expression (MWE), that is a cohesive lexeme that crosses word boundaries (Sag, Baldwin, Bond, Copestake, & Flickinger, 2002). As we shall discuss, knowledge of this type of collocation is important for dictionary construction, language analysis and generation but high precision extraction from corpora is generally more difficult and is the focus of much current research (McCarthy, Villavicencio, Korhonen, & Bond, 2003).

Some researchers (McCarthy, Keller, & Carroll, 2003; Bannard, Baldwin, & Lascarides, 2003; Baldwin, Bannard, Tanaka, & Widdows, 2003) distinguish between compositional and non-compositional collocations. If a collocation is compositional, its meaning can be determined from its constituent parts. What makes these expressions true collocations, rather than just frequently occurring word combinations, is that they are institutionalised in the language. As discussed by Pearce (2001), an alternative expression that has the same compositional semantics would not be used by a native speaker. For example, in English we say "strong tea" not "powerful tea" (Halliday, 1966), and "emotional baggage" not "emotional luggage" (Lin, 1998b). The first expression in each case is a collocation and the second is an anti-collocation (Pearce, 2001). Compositional collocations tend not to pose a problem for natural language understanding systems since the meaning can be established from the parts. However, it is a problem for foreign language learners

or, more generally, generation systems, which need to be able to choose between the alternatives. Non-compositional collocations e.g. "hot dog", where the meaning cannot be established from the constituent parts, pose a problem both in understanding and in generation. Such collocations need to be listed in the lexicon and are therefore also of interest to lexicographers.

Pearce (2002) and Lin (1998b) both propose techniques for collocation extraction based on this idea of non-substitutability. For example, Pearce substitutes each word in a potential collocation with synonyms in WordNet and compares the MLE probabilities of each expression with the expected probability of the expression if the words occurred independently. Collocations occur more frequently than expected given their constituent parts and anti-collocations occur less frequently than expected given their constituent parts. Lin (1999) substitutes each word in the potential collocation with the ten most similar words according to an automatically generated thesaurus (Lin, 1998a) and detects a collocation as a significant difference in mutual information. Such a technique views a collocation as occurring where the distributional hypothesis breaks down: where words that can normally be substituted for each other, cannot be substituted for each other. The success of this technique also demonstrates that collocations should be an important consideration in similarity-based language modelling techniques.

Baldwin et al. (2003) recognises that although these substitutability techniques identify institutionalised collocations, they are not a sufficient test for compositionality. As discussed by Bannard et al. (2003), the meaning of a compositional phrase implies the meaning of its constituents. For example, "strong coffee" implies *coffee* and *strong* and when the phrasal verb "to put up" takes a direct object there is an implication that something is being *put* somewhere and something is *up*. On the other hand, the phrasal verb "to carry out (a task)" is non-compositional because nothing is being *carried* and nothing is *out*. In the compositional cases, the collocation can be considered to be a hyponym of its head, whereas in the non-compositional cases, it is not (Haspelmath, 2002). This idea is used by Baldwin et al. (2003), who consider hyponymy and semantic distance in WordNet between a collocation and its head. An alternative approach, taken by McCarthy et al. (2003), is to consider the distributional similarity between a collocation and its head. The hypothesis here is that a non-compositional collocation will have different selectional preferences to its head. Alternatively, we can say that if a collocation is a hyponym of its head (i.e. it is compositional), we should be able to substitute the collocation with its head.

2.2.13 Text Simplification

The aim of the Practical Simplification of English Text (PSET) project (Carroll, Minnen, Canning, Devlin, & Tait, 1998) was to automatically simplify text for aphasic readers. Aphasic readers typically have problems with complex syntactic constructions, long sentences and also with rare words (Devlin, 1999). The system's lexical simplifier identifies synonyms of content words using WordNet and interrogates the Oxford Psycholinguistic Database (Quinlan, 1992) to obtain the respective frequencies of the original word and its synonyms. The word with the highest frequency is selected and used in the simplified text.

There are two major potential problems in this approach to lexical simplification. The first is word sense ambiguity. However, Carroll et al. (1998) note that, in practice, most candidate words for simplification often have only a single sense. Ambiguous words tend to be more frequent and

it is therefore less likely that the system will find a more frequent synonym. The second problem is the identification of collocations, since, as discussed in the previous section, it is not generally appropriate to substitute a word with one of its synonyms within the context of a collocation (Pearce, 2001, 2002).

However, these problems aside, this is an obvious potential application for an automatically generated thesaurus, which focuses on the substitutability of words. If distributional similarity is an accurate predictor of semantic similarity, then distributional information alone might be enough to select an appropriate substitute. If not, distributional similarity could be used to eliminate synonyms which tend to be used in different contexts.

2.2.14 Compound Noun Interpretation

In the interpretation of compound nouns, it is necessary to determine the implicit relationship between the two nouns. For example, "terrorist activities" refers to activities *of* terrorists whereas "gun crime" refers to crime *with* guns. One approach to this problem (Lauer, 1996) is to compare the trigram frequencies of the potential paraphrases. For example, which of "activities of terrorists", "activities with terrorists" and "activities for terrorists", occurs more frequently in text? However, this approach obviously suffers from the data sparseness problem. To tackle this, we can, as in other applications, use a thesaurus and establish the relationships probabilistically over semantic classes rather than individual lexical items.

2.2.15 Topic Identification

The first step in many proposed text summarization systems is the extraction of key concepts from the source text (Silber & McCoy, 2002). One approach, known as lexical chaining (Morris & Hirst, 1991), exploits the property of text cohesiveness, that is texts tend to contain many words pertaining to the main topics or themes of the text. Since these words are all related to the same topic, they will also tend to be inter-related and hence, by identifying long chains of semantically related words running through the text, the key concepts in the text can be identified. Such a system obviously requires a source of semantic similarity information, which could be an automatically generated thesaurus.

2.2.16 Spelling Correction

The correction of real-word spelling errors, i.e. where two words are confused or one is mistyped in place of the other, is another application of similarity-based methods. In general, the task is to choose which of a set of (easily-confused) alternatives is the most likely given the context. There are two main approaches (Yarowsky, 1994; Golding, 1995) in the literature, which we now discuss.

The first approach casts the problem as a lexical disambiguation task and tests for collocations or patterns of words around the target word. For example, a context-sensitive spelling corrector might hypothesize that the word *principal* was intended in the phrase "the school principle". By analogy with word sense disambiguation there are a number of different ways these contextual patterns can be learnt. For example, Mangu and Brill (1997) and Brill (2000) apply transformation-based learning to the problem. Alternatively, co-occurrence probability estimates

for each alternative can be derived from corpus data, which means this approach reduces to the language modelling task described earlier. For this reason, in this work, we focus on the second approach, which is semantic and which we now discuss.

The second approach (Hirst & St-Onge, 1998) looks at more global context and exploits the property of text cohesiveness, that is, naturally occurring coherent texts tend to contain many semantically related words (Morris & Hirst, 1991). Budanitsky and Hirst (2001) exploit this property by assuming that the author's intended word is more likely to be semantically related to the rest of the text than the real-word spelling error.

2.2.17 Clustering

Many of the applications considered here potentially involve forming clusters of similar words. There is a large body of work on clustering algorithms (Manning and Schütze (1999) provides an introduction) and, in general, the clustering algorithm can be chosen independently of the distance or similarity measure. Accordingly, it should be appreciated that this thesis is a study of the similarity measures which might be used within a particular clustering technique rather than a study of different clustering techniques. For the purposes of illustration, we will now present three of the clustering techniques used in the NLP literature (Pereira et al., 1993; Hofmann & Puzicha, 1998; Schütze, 1998; Donghong, Jun, & Changning, 1998; Lee & Pereira, 1999; Rooth, Riezler, Prescher, Carroll, & Beil, 1999; Clark, 2000; Lin & Pantel, 2001; Dhillon, Mallela, & Kumar, 2002; Li, 2002; Schulte Im Walde & Brew, 2002; Korhonen, Krymolowski, & Marx, 2003) and then discuss the *k-nearest neighbours* technique which is used throughout this thesis.

Agglomerative hierarchical clustering (Cutting, Karger, Pedersen, & Tukey, 1992) is initialised by assigning each object to its own cluster. The two closest (or most similar) clusters are determined and merged. This is repeated until the required number of clusters is obtained. *Nearest neighbour* clustering is an example of a graph-theoretic method (van Rijsbergen, 1979) where an object is connected to its nearest neighbour (i.e. the most similar other object) or, alternatively, to all objects with which it has a similarity above a certain threshold. Clusters are then formed by finding all of the connected components. In *k-means* clustering (Forgy, 1965), objects are assigned to k initial clusters either randomly or using some other clustering method. The distance from each object to the centroid of each cluster is computed and objects are reassigned to their nearest cluster. This process iterates until there are no more reassignments.

All of the afore-mentioned techniques are hard clustering techniques since each object can occur only in a single cluster. If the cluster representation of object A is B,C,D,E, then the cluster representation of D is A,B,C,E. This type of clustering technique implicitly assumes that similarity between objects is symmetric. An asymmetric measure can be used, since one can compute the similarity from the object to the centre of each proposed cluster. However, the final cluster formation assumes that if A is to be used in the representation of B then B is to be used in the representation of A.

The *k-nearest neighbours* technique (Lee, 1999) used in this thesis can be viewed as a soft clustering technique where each object is the centre of its own cluster. Each object's cluster is made up of its k nearest neighbours. For example, the cluster representation of A might be B,C,D,E whilst the cluster representation for D is B,C,E,F. We use this technique because it is analogous to cre-

ating a thesaurus entry for each word, it does not assume that similarity is symmetric, it is simple to implement and it provides a fairly transparent way of evaluating similarity methods. The only parameter that needs to be considered, other than the similarity measure, is the neighbourhood size k and we explore whether this can be optimised globally or whether it is a word-dependent property. k-nearest neighbours techniques have also been used in pseudo-disambiguation experiments (Lee, 1999) and *distance-weighted averaging* (Grishman & Sterling, 1994; Dagan et al., 1999; Lapata, Keller, & McDonald, 2001). The latter is a smoothing technique that computes a distance-weighted average of co-occurrence probability estimates from an object's k nearest neighbours.

2.3 Evaluation of Distributional Similarity Measures

In this section, we consider four standard approaches to evaluation and their applicability in the area of distributional similarity. These approaches are: evaluation against a gold standard, evaluation against human plausibility judgements, evaluation against or prediction of unseen data and application-based evaluation tasks. We will then consider two actual evaluation tasks used in the literature. These are pseudo-disambiguation and a WordNet prediction task. Variations of these two tasks are used throughout this thesis.

2.3.1 Evaluation Against a Gold Standard

Perhaps, the simplest approach to evaluation in NLP is to use a gold standard. One simply compares the data or answers provided by the technique being evaluated with the correct answers as given by the gold standard. For example, in parsing one might parse the text of the Penn Treebank (Marcus, Santorini, & Marcinkiewicz, 1993) and then compare the trees that have been produced with the gold standard trees of the Penn Treebank. For example, Collins (1999) and Charniak (2000) train their statistical parsers on a portion of the Penn Treebank and test them by parsing sentences from a held-out, unseen, portion. In the field of word sense disambiguation, one might use sense-tagged data hand-annotated by a team of lexicographers as a gold standard against which to evaluate automatic sense disambiguation techniques. For example, SemCor (Miller, Chodorow, Landes, Leacock, & Thomas, 1994) is a portion of the Brown corpus hand-tagged with WordNet senses. This, however, may not be appropriate if one wishes to use a different sense inventory. In this situation, one inexpensive way of acquiring a partially sense-tagged corpus is to construct one from examples of sense usages in dictionary definitions (Preiss, 2000; Weeds, 2000).

However, there are three main issues to consider when considering using a gold standard in evaluation.

First, is there a readily available gold standard such as the Penn Treebank? If not, it can be time-consuming and therefore expensive to construct one. An automatic sense disambiguation technique might run on the BNC in less than a day, but it would take a human lexicographer many months or even years to create an equivalent gold standard.

Second, if there is a gold standard, do we actually need the automatic technique under development? Why not just use the gold standard? Obviously, the answer to this in many cases is that there is only a partial gold standard. The Penn Treebank only contains correct parses for the sentences in the Penn Treebank and human annotators can only annotate a finite amount of text

with sense tags. The idea of using a gold standard is that if our automatic technique works well on that data, we might assume that it will work well on other data for which we don't have a gold standard. However, this can be a dangerous assumption. Results for a particular domain or genre may not be generalizable to other domains and genres.

Third, again if there is a gold standard, how valid is it? Are there likely to be any errors in it or are we working in a field where there might be more than one correct answer? Linguists argue over the correct parse for a sentence and lexicographers argue over the sense assigned to a particular word in context, indeed they argue over the senses that a word should have. If our automatic technique suggests something different to the gold standard, is it necessarily wrong? Penn Treebank parse trees can only be used as a gold standard if we want to produce parse trees that look like Penn Treebank parse trees. Maybe, we want our technique to learn something different. For example, we may want to produce a deeper level of analysis than contained in Penn Treebank trees.

These issues are very real problems in the area of lexical semantics. For example, Pearce (2002) evaluates collocation extraction techniques against a gold standard, constructed from multi-word information in the machine-readable version of the New Oxford Dictionary of English (Pearsall & Hanks, 1998). However, as Pearce comments, "it is somewhat controversial to evaluate these techniques with reference to anything considered as a standard." One might ask whether the standard contains all the collocations in current use, whether it contains any which have fallen into disuse, and whether, if it really is the perfect gold standard, whether there is any need to automatically extract collocations.

The use of a gold standard in the evaluation of distributional similarity measures is even more problematic since there is no gold standard where we can look up a word's set of distributionally nearest neighbours, let alone a gold standard containing the exact distributional similarity scores between each pair of words. Accordingly, if we want to use a gold standard, we either have to find a team of distributional similarity experts to construct one or we have to cheat and use something which is not really a gold standard for distributional similarity. The former option is not overly-appealing due to its time-consuming nature and the fact that it assumes the existence of distributional similarity experts (something which we will come back to in Section 2.3.2), which leaves only the latter option.

The much discussed link between semantics and distributional characteristics of a word, has led to semantic gold standards being used in the evaluation of distributional similarity techniques (Lin, 1998a; Curran & Moens, 2002a). Further, if the aim of distributional similarity techniques is to automatically generate thesauruses, this would seem a perfectly reasonable thing to do. However, this still leaves the second two issues raised above unresolved. If we already have a gold standard for semantic similarity, do we need to automatically generate semantic similarity data? The answer to this is yes. Assuming the results are generalizable, we can apply the automatic techniques to other languages or in particular domains or genres. Further, since there are always new words coming into a language, the techniques could be used to extend existing resources. However, this issue of language change leads us to the third issue: is the gold standard valid? As well as new words coming into a language, words change in their meaning and use. This may suggest that distributional techniques could be used to study this change. However, it also casts

doubt on the use of a fairly static gold standard in evaluation. Can we be sure that it captures every semantic link between words as they are currently used? If our automatic techniques tell us that two words, unrelated in the gold standard, have same degree of semantic similarity, is it necessarily wrong? Is it not possible, that we have learnt a new semantic linking, which has been previously unobserved?

Further, as discussed in the previous section, there are potential applications for distributional similarity techniques which do not necessarily require the words to be semantically similar as well as distributionally similar. We may be doing badly on the evaluation technique not because of the distributional similarity technique used but because of a flaw in the hypothesis which links distribution and semantics. Accordingly, we still require an evaluation technique which evaluates distributional similarity independent of semantic similarity.

2.3.2 Evaluation Against Human Plausibility Judgements

Evaluation against human plausibility judgements is related to evaluation against a gold standard and is often used when a gold standard is unavailable or incomplete.

This alternative to using a gold standard is often used in the field of collocation extraction (Lapata, McDonald, & Keller, 1999; Lapata et al., 2001; Blaheta & Johnson, 2001). A set of potential collocations, e.g. adjective-noun pairs, are constructed and each member of a group of humans rates how plausible each is as a collocation. The judgements of the automatic technique can then be correlated against the human judgements. A similar approach was used by Lapata and Lascarides (2003) to evaluate a metonymy resolution algorithm. Potential paraphrases of metonymic phrases were generated and human subjects asked to rate their plausibility. For example, the phrase "finished his dinner" might be interpreted, amongst other things, as meaning "finished cooking his dinner" or "finished eating his dinner". In general, we would expect humans to favour the latter interpretation and hence an automatic technique that favours this interpretation would be rated higher.

Another area in which human plausibility judgements have been used is in the evaluation of semantic similarity measures. Much of this work is based on human synonymy judgements obtained by Rubenstein and Goodenough (1965) on 65 pairs of words and/or human synonymy judgements obtained by Miller and Charles (1991) on a subset of 30 pairs of words from the Rubenstein and Goodenough set. These data sets have been used by Resnik (1995), Jiang and Conrath (1997), Lin (1998c), Budanitsky and Hirst (2001) and Jarmasz and Szpakowicz (2003) to evaluate a number of different semantic similarity measures based on thesauruses such as WordNet and Roget.

One problem with using human plausibility judgements is the subjectivity of the judgements. The subjectivity is of course reduced by using a group of humans and can be further reduced by discarding test instances where there is a large amount of disagreement. However, whether subjectivity and priming effects can be completely eliminated, is not a question which we will attempt to answer here, since there is another, more fundamental, problem connected to using human plausibility judgements in the evaluation of distributional similarity measures.

The fundamental problem is whether it is appropriate to use such an evaluation technique in the field of distributional similarity. Can a human judge how plausible it is that two words are

distributionally similar? Being able to decide which of two pairs of words are more distributionally similar to each other is a more difficult task than being able to decide which of two collocations is more plausible. The plausibility of collocations is simple to decide because it is part of the psychological process of lexical choice, which we use all the time. If we are asked, "Which is more plausible: *strong coffee* or *powerful coffee*", we can simply think, "Which would I be more likely to say?" However, deciding which words appear in more similar contexts, would not appear to be part of any simple psychological process. Deciding which of two pairs of words, e.g. *(coffee, tea)* and *(coffee, water)*, is more plausible as a pair of distributionally similar words involves considering every different context in which each word occurs. Further, it is hard to imagine a human being able to completely tease apart their intuitive notions about distributional and semantic similarity. For example, how many people would rate the pair *(dog, road)* over *(cat, street)* on the basis that *dog* and *road* are both nouns which can occur as the object of the verb *walk*? Accordingly, it would seem that any use of human plausibility judgements in the area of distributional similarity must be in the context of an application of distributional similarity. We will discuss the use of application-based tasks in Section 2.3.4.

2.3.3 Evaluation Against Unseen Data

Our next evaluation approach comes from the domain of language modelling. A good language model will assign higher probabilities to sentences, or texts, taken from the language than to strings of words which do not form part of the language. Further, a good language model must be able to assign a probability to a string which is an actual sentence in the language even if it has never before been uttered. For example, the sentence quoted earlier from Jane Austen's Emma, had probably never been uttered before Jane Austen wrote it; yet, even at the time, it would have been greatly more probable than nonsensical or ungrammatical utterances such as either of Chomsky's now oft-quoted linguistic examples of "Colourless green ideas sleep furiously" and "Furiously sleep ideas green colourless." (Chomsky, 1964).

Accordingly, many language models are evaluated on their ability to predict a set of "held-out" or "unseen" test sentences. For example, given a data set of 100000 sentences, one could build language models based on 80000 of them and evaluate the language models on the basis of the probabilities they assign to the remaining 20000.

Two standard language model evaluation metrics based on this intuition are test set perplexity and relative entropy of the test set given the model. As discussed by (Cover & Thomas, 1991), relative entropy measures the inefficiency, in bits, of assuming the model distribution, q, when the true distribution of events is the distribution observed in the test set, p:

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{2.1}$$

Test set perplexity can be defined (Chen & Goodman, 1998; Rosenfeld, 2000) as a function of the related measure of cross-entropy:

$$perplexity(p;q) = 2^{E_c(p;q)} \tag{2.2}$$

where cross-entropy E_c is given by:

$$E_c(p;q) = -\sum_x p(x)\log q(x) \tag{2.3}$$

Such an evaluation approach has already been applied in the area of distributional similarity methods. Dagan et al. (1999) smooth co-occurrence probabilities using the distributions of the distributionally nearest neighbours (as computed using relative entropy). Test set perplexity is then used to compare the similarity-based language model with a back-off language model.

There are, however, three main problems with this evaluation approach. First, the observed, empirical distribution in the held-out test data will never be the population distribution. Hence, however good our language model, it will never model the test set distribution exactly. In fact, it should not model the test set distribution exactly, since in this case it would be an inefficient model for a different test set. Hence, the approach can only be used comparatively and it is hard to place a lower bound on what relative entropy or test set perplexity we could expect to achieve.

Second, the performance of the language model depends both on the distributional similarity measure used and the algorithm used to smooth distributions with those of their neighbours. This makes it difficult to evaluate the similarity measures independent of the smoothing algorithm, since there might be positive or negative interactions between certain algorithms and measures.

Third, there is a whiff of circularity in using a measure such as relative entropy to evaluate measures of distributional similarity that include relative entropy. In both creating and evaluating the language model, we are concerned with the similarity between probability distributions. In the former, it is the similarity between lexical co-occurrence distributions. In the latter, it is the similarity between a model distribution and an empirical distribution. However, one might think that if there are different ways of measuring one which give different results, there might be different ways of measuring the other which also give different results. In using such an evaluation approach, we are implicitly assuming that we have a good measure of similarity for two probability distributions. This then leads us to ask why, in that case, do we need to consider other measures when the probability distributions are based on lexical co-occurrences?

2.3.4 Application-Based Evaluation Tasks

The fourth evaluation approach considered here is the use of application-based tasks. Rather than attempting to evaluate the similarity results directly, one can evaluate how useful they are in one of the many applications discussed earlier. Indeed, we have already considered two such evaluation tasks. Evaluation against a semantic gold standard is an evaluation task in the application of automatic thesaurus generation and evaluating against unseen data is an evaluation task for the application of language modelling. However, now we can envisage an evaluation task corresponding to every application mentioned in Section 2.2. Simply implement an application and see which measure of distributional similarity gives the best results in terms of performance in that application.

However, now we have to consider whether high performance in one application is correlated with high performance in another. For example, are the most useful neighbours for word sense disambiguation also the most useful neighbours for prepositional phrase ambiguity resolution? Unfortunately, it is not possible to consider every application separately in this thesis. However, the results we do have from just two applications suggest that semantic applications may require

something quite different of a word's nearest neighbours to applications which might be considered to be in the distributional or syntactic domain.

We now consider two specific application-based evaluation tasks which have been used in the literature. First, we will consider pseudo-disambiguation as an evaluation technique for language modelling and word sense disambiguation. We will then consider a WordNet prediction task as an evaluation technique for automatic thesaurus generation.

2.3.5 Pseudo-Disambiguation Tasks

Pseudo-disambiguation tasks have become a standard evaluation technique (Gale, Church, & Yarowsky, 1992; Schütze, 1992b; Pereira et al., 1993; Schütze, 1998; Lee, 1999; Dagan et al., 1999; Golding & Roth, 1999; Rooth et al., 1999; Even-Zohar & Roth, 2000; Lee, 2001; Clark & Weir, 2002) and, in the current setting, we may use a noun's neighbours to decide which of two co-occurrences is the most likely. Although pseudo-disambiguation itself is an artificial task, it has relevance in at least two application areas. First, by replacing occurrences of a particular word in a test suite with a pair or set of words from which a technique must choose, we recreate a simplified version of the word sense disambiguation task; that is, choosing between a fixed number of homonyms based on local context. The second is in language modelling where we wish to estimate the probabilities of co-occurrences of events but, due to the sparse data problem, it is often the case that a possible co-occurrence has not been seen in the training data.

A typical approach to performing pseudo-disambiguation is as follows. A large set of noun-verb direct-object pairs are extracted from a (parsed) corpus. A portion of the data is used as test data and another portion is used as training data. The training data can be used to construct a language model and/or determine the distributionally nearest neighbours of each noun. Noun-verb pairs (n, v_1) in the test data are replaced with noun-verb-verb triples (n, v_1, v_2) and the task is to decide which of the two verbs is the most likely to take the noun as its direct object. Performance is usually measured as error rate.

$$\text{Error rate} = \frac{1}{T}\left(\#\text{ of incorrect choices} + \frac{\#\text{ of ties}}{2}\right) \tag{2.4}$$

where T is the number of test instances and a tie results when it is not possible to decide between the alternatives.

Approaches can vary, however, in the details of constructing the training and test sets and also the way in which the decision between alternatives is made. This will now be discussed.

One approach (e.g. Pereira et al. (1993), Clark and Weir (2002)) is to form the test set by randomly selecting a percentage of the noun-verb pairs from the original data set, ensuring that each verb occurs sufficiently frequently. The training data is then formed by deleting each instance of a test noun-verb pair from the original data. This is to ensure that the test data is completely unseen. The test set is then completed by selecting the verb v_2 such that it has approximately the same frequency as v_1 in the original data. Clark and Weir (2002) also ensure that (n, v_2) does not occur in the training data. This ensures that (n, v_1) is more probable than (n, v_2) and that any technique which relies on MLE probabilities or backs-off to unigram probabilities (Katz, 1987) will perform no better than chance. The techniques of Pereira et al. (1993) and Clark and Weir (2002) use the class or cluster of nouns containing the target noun n to decide which is the most

probable co-occurrence. Pereira et al. (1993) find the clusters of (distributionally) similar nouns using the training data and each member of the target noun's cluster votes for the verb with which it co-occurs most frequently in the training data. Clark and Weir (2002) use semantic classes derived from WordNet. Counts of verb co-occurrences are accumulated up the WordNet noun hyponym hierarchy and the appropriate level of generalisation over the target noun is determined using χ-squared tests.

A different approach to test set construction (Lee, 1999; Dagan et al., 1999; Lee, 2001) is to randomly select a portion of the co-occurrences as test data and then delete any instances of co-occurrences from this test data which also occur in the training data. We feel that this method of test set construction is more appropriate for a large-scale evaluation of distributional similarity measures, since completely removing co-occurrence types from the training data (to ensure that they are unseen) distorts the distributional data used by the measures to determine similarity. This alternative approach ensures that test instances are unseen whilst not distorting the training data.

The approach of Even-Zohar and Roth (2000) differs in that they do not ensure that the test instances are unseen. This may explain why they achieve a much higher reduction in error rate over the baseline than Lee (1999). The arguments for and against ensuring the test data is unseen will be examined in Chapter 5.

Another possible difference in the method of test set construction is whether the *wrong* choice verb, v_2 is required to be unseen with the noun, n, in the training data. If (n, v_2) can have occurred in the training data, then we cannot strictly say that the first verb v_1 is the *correct* choice verb "by construction" of the test set. However, in practice (Lee & Pereira, 1999), it seems that ensuring that (n, v_2) is unseen in the training data makes little difference to the overall pattern of results.

The work of Lee (1999), Dagan et al. (1999) and Lee (2001) also differs from that of Pereira et al. (1993) in how the decision between alternatives is made. Pereira et al. (1993) form clusters of distributionally similar words and each noun in the appropriate cluster is given a vote as to which verb is preferred. In the later work, the k nearest neighbours of the noun, n, are given a vote as to which is the preferred verb, where k is a parameter which can be optimised. The potential advantages of each approach are discussed by Lee and Pereira (1999). We would expect the k nearest neighbours approach to perform better because each noun is at the centre of its own cluster. Optimising k as a parameter means that we do not have to fix the cluster size in advance. In general, we would expect error rate to initially decrease as more neighbours are considered, since more decisions will be able to be made. However, after a certain optimal number of neighbours, error rate will start to increase again because the neighbours are no longer similar enough to make the correct decision.

The paradigm we follow in our initial experiments is more closely aligned to that of Lee (1999) than the other approaches. The differences are that we ensure that verb v_2 has not occurred with n in the training data and we implement a voting scheme where each of the k nearest neighbours is given a vote which is proportional to the difference in its frequency of co-occurrence with each of the two verbs. This means that neighbours which co-occur with each verb with an approximately equal probability have less influence on the voting process than neighbours which occur with one verb significantly more than the other. However, these issues, and more, will be returned to in Chapter 5.

2.3.6 WordNet Prediction Task

The WordNet prediction task used by Lin (1998a) evaluates the ability of a distributional similarity measure to predict similarity between words as defined by a measure based on the hyponymy relation in WordNet (version 1.6). This task evaluates the usefulness of a distributional similarity measure as a predictor of semantic similarity and therefore its potential for use in automatic thesaurus generation. An underlying assumption is that the hyponymy relation in WordNet is a gold standard for semantic similarity which is, of course, not true. However, it can be argued that a distributional similarity measure which more closely predicts WordNet is more likely to be a good predictor of semantic similarity. We will now consider the approach taken by Lin in more detail.

First, it is necessary to consider that there is more than one way to measure the distance between two nouns in the WordNet noun hierarchy. These range from simple edge-counting techniques (Rada, Mili, Bicknell, & Blettner, 1989) to techniques which consider link direction (Leacock & Chodorow, 1998) to techniques which factor in corpus probabilities (Jiang & Conrath, 1997). See Budanitsky (1999) for an extensive survey of WordNet-based similarity measures.

The WordNet-based similarity measure used by Lin (1998a) is based on a proposal in Lin (1997):

$$sim_{wn}(n_1,n_2) = \max_{c_1 \in S(n_1) \wedge c_2 \in S(n_2)} \left(\max_{c \in sup(c_1) \cap sup(c_2)} \frac{2\log P(c)}{\log(P(c_1)) + \log(P(c_2))} \right) \tag{2.5}$$

where $S(n)$ is the set of senses of the noun n in WordNet and sup(c) is the set of possibly indirect super-classes of concept c in WordNet. This measure is an implementation of Lin's similarity theorem (Lin, 1998c) which states that the similarity between objects A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are. In the current context, the two objects are nouns, the commonality of two concepts is the maximally specific common super-class of those concepts and the similarity between two nouns is defined as the similarity between their most similar senses. The probability $P(c)$ that a randomly selected noun refers to an instance of concept c, is estimated by the frequency of concepts in SemCor (Miller et al., 1994), a sense-tagged subset of the Brown corpus. An occurrence of a concept (i.e. word sense) in SemCor refers to instances of all of the super-classes of that concept. For example, if we see the *meat* sense of the word *chicken*, we have seen instances of *chicken*$_{meat}$, *meat*, *food* etc. From a practical point of view, this scheme is necessary to ensure that a super-class is more probable (and therefore contains less information) than any of its children.

Lin (1998a) computes the 200 nearest neighbours according to each similarity measure for a set of 4294 nouns. He then computes the similarity, or overlap, between each pair of neighbour sets for a given noun. The similarity between two neighbour sets, or thesaurus entries, for the same word is calculated as follows. Suppose entries are constructed for noun n using measures A and B. The k nearest neighbours and their associated similarities according to each measure might be written:

$$neighs(n,A,k) = \{(n_{A_1},s_{A_1}),(n_{A_2},s_{A_2}),...,(n_{A_k},s_{A_k})\}$$

$$neighs(n,B,k) = \{(n_{B_1},s_{B_1}),(n_{B_2},s_{B_2}),...,(n_{B_k},s_{B_k})\}$$

The similarity between these sets can be computed as:

$$sim\left(neighs(n,A,k),neighs(n,B,k)\right) = \frac{\sum_{n_{A_i}=n_{B_j}} s_{A_i} s_{B_j}}{\sqrt{\left(\sum_{i=1}^{k} s_{A_i}^2\right)\left(\sum_{j=1}^{k} s_{B_j}^2\right)}} \tag{2.6}$$

This score is 1 if and only if the neighbour sets are identical, that is they contain exactly the same neighbours with exactly the same similarity scores. At the opposite end of the scale, the score is 0 if the sets do not share any neighbours.

Having computed the similarity between each pair of neighbour sets for 4294 nouns, Lin computes the mean similarity between the thesaurus entries and the standard error of the mean, defined as the standard deviation of the data items divided by the square root of the number of data items.

More recently, the work of James Curran (Curran & Moens, 2002a, 2002b; Curran, 2002), has compared the neighbour sets produced by distributional measures with thesaurus entries derived from thesauruses other than WordNet (the MacQuarie (Bernard, 1990), Moby (Ward, 1996) and Roget (Roget, 1911)). However, as discussed earlier, if grammatical relation data is used as context for the distributional similarity measures, we expect the neighbour sets produced to be more akin to those derived from the hyponymy relation in WordNet than Roget (Lin, 1998a; Kilgarriff & Yallop, 2000). One argument for using sources other than the noun hierarchy in WordNet could be that distributional similarity measures based on the concept of lexical substitutability also return pairs of words related by the antonymy relation. However, whilst a measure of semantic similarity derived from the WordNet noun hierarchy does not explicitly consider the antonymy relation, most pairs of antonyms are also hyponyms of the same common super-class. Consequently, antonyms are still found as semantically related words.

2.4 Conclusions

In this chapter, we have considered the intuitions behind distributional similarity. We have discussed what it means for two words to be distributionally similar and the different orders of affinity which exist between words. We follow work using grammatical relation data to define context. Defining context syntactically means that distributional similarity can be thought of in terms of lexical substitutability.

Further, we have discussed a large number of potential applications of knowing distributionally similar or lexically substitutable words. Some of these applications take advantage of the much-discussed link between lexical substitutability and semantic relatedness. Other applications, such as probability estimation techniques, make use of the property of distributional similarity directly.

Evaluation is a major problem in this area of research. We argue that humans are unable to judge lexical distributional similarity independent of semantic similarity and that, in any case, there is no single "right answer" as to which words are most distributionally similar. The best measure of distributional similarity will be the one which returns the most useful neighbours in the context of a particular application and thus leads to the best performance in that application. This thesis investigates whether high performance in one application area is correlated with high performance in another. If not, the best choice of distributional similarity measure depends on

the application. In the next chapter, we begin our investigation by introducing a large, varied collection of existing distributional similarity measures.

Chapter 3

Existing Measures of Distributional Similarity

Over the years a broad range of measures have been proposed or adopted for measuring lexical distributional similarity. These measures originate from fields such as geometry, statistics, probability theory, information theory and information retrieval (IR). In this chapter, we present the theoretical basis for fifteen existing measures. We then compare these measures in terms of the neighbour sets that they produce from the same data and examine their tendencies to select high or low frequency words as neighbours.

3.1 Theory

Throughout our discussion we will consider similarity between nouns based on their co-occurrences with verbs in the direct-object relation. The *co-occurrence types* of a noun are verbs and these have associated frequencies which may be used to form probability estimates. The noun for which we are finding neighbours will be referred to as the *target noun* or *target word*. If we are computing the similarity between the target noun and another noun, then the second noun is a *potential neighbour* of the target noun. A target noun's *nearest neighbours* are the potential neighbours which have the highest similarity with the target noun.

Our use of the term similarity measure encompasses measures which should strictly be referred to as *distance*, *divergence* or *dissimilarity* measures. An increase in distance correlates with a decrease in similarity. This distinction is relevant if we wish to weight neighbours by their similarity. In this case, a distance must be converted into a similarity score, using a function such as the one proposed in Dagan et al. (1999):

$$sim(w_1, w_2) = 10^{-\beta dist(w_1, w_2)} \tag{3.1}$$

where β ranges from 0 to $+\infty$ and controls the relative influence of the closest neighbours. In general, β must be less than 1 for distant neighbours to have non-negligible similarities with the target word. Other functions to convert between distance and similarity are possible and these may yield different results in a particular application. For this reason, our evaluation will focus on the ranks of neighbours rather than the particular similarity scores given to individual word pairs. In

this case, the distinction between similarity and distance is irrelevant since both types of measure can be used to determine the nearest neighbours of a word.

We will now consider each measure or family of measures in turn. We consider geometric measures (the Minkowski Distance (L_1 and L_2 Norms) and the cosine measure) in Section 3.1.1, correlation coefficients in Section 3.1.2, combinatorial measures (Jaccard's Coefficient and the Dice Coefficient) in Section 3.1.3, substitutability measures (based on the Kullback-Leibler Divergence Measure or the confusion probability) in Section 3.1.4 and mutual information (MI) based measures (Hindle's Measure and Lin's Measure) in Section 3.1.5.

3.1.1 Geometric Measures

Geometric measures are based on vector geometry. The co-occurrence types are the dimensions of the vector space and the frequencies of co-occurrence of the target word with each co-occurrence type are used to calculate the value of each component of the vector. In general, the value of each dimension's component is the conditional probability of the co-occurrence type given the target word. For example, if we are considering the similarity between two nouns based on co-occurrence with only two verbs, for the sake of simplicity, the co-occurrence information in Table 3.1 can be represented by the 2-d vector diagram in Figure 3.1.

	Frequencies			$P(verb\|noun)$	
	see	*get*	Total	*see*	*get*
bus	9	68	77	0.12	0.88
dog	43	24	67	0.64	0.36
Total	52	92	144		

Table 3.1: Co-occurrence information for *bus* and *dog*

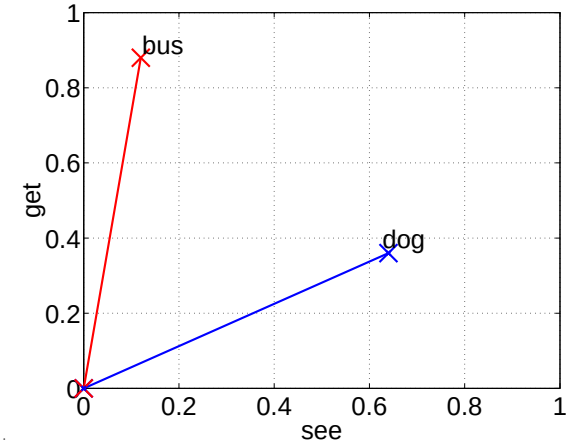

Figure 3.1: 2-d vector diagram for *bus* and *dog*

3.1.1.1 The Minkowski Distance

The Minkowski Distance is a family of metrics that can be used to measure the distance between two points in space. Since our dimensions are verbs and the values of our vector components are

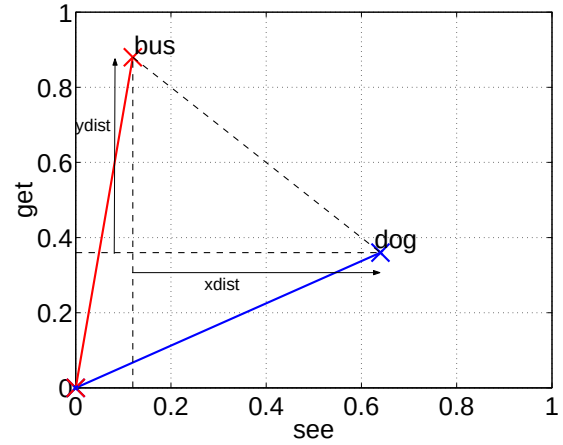

Figure 3.2: Geometric distance metrics

the probabilities of the verbs given the noun, the Minkowski Distance between two nouns, n_1 and n_2 can be written as:

$$dist_{mink}(n_1,n_2) = \sqrt[n]{\sum_v |P(v|n_1) - P(v|n_2)|^n} \tag{3.2}$$

where n is a positive integer. The use of $n = 1$ leads to the familiar L_1 Norm:

$$dist_{L_1}(n_1,n_2) = \sum_v |P(v|n_1) - P(v|n_2)| \tag{3.3}$$

The L_1 Norm is also known as the Manhattan Distance, the taxi-cab distance, the city-block distance and the absolute value distance, since it represents the distance travelled between the two points if you can only travel in orthogonal directions (a common restriction in many US cities). In our 2-d example, see Figure 3.2, the value of the L_1 Norm is $xdist + ydist$. The L_1 Norm, although simple, has been shown to be as effective as more complicated similarity measures (Lee, 1999) and is widely used in clustering (Kaufman & Rousseeuw, 1990; Cutting et al., 1992; Schütze, 1993; Dagan et al., 1999). The L_1 Norm can be computed efficiently since it can be rewritten (Dagan et al., 1999) as:

$$dist_{L_1}(n_1,n_2) = 2 - \sum_{v \in V(n_1) \bigcap V(n_2)} P(v|n_1) + P(v|n_2) - |P(v|n_1) - P(v|n_2)| \tag{3.4}$$

where $v \in V(n)$ iff $P(v|n) > 0$. It follows from the triangle inequality that $0 \leq L_1(n_1,n_2) \leq 2$, with equality to 0 if and only if $P(v|n_1) = P(v|n_2)$ for all v, and equality to 2 if and only if there are no verbs v for which both $P(v|n_1) > 0$ and $P(v|n_2) > 0$.

The other widely used parameter setting in the Minkowski Distance is $n = 2$. This gives us the L_2 Norm (also known as the Euclidean Distance):

$$dist_{L_2}(n_1,n_2) = \sqrt{\sum_v (P(v|n_1) - P(v|n_2))^2} \tag{3.5}$$

In our 2-d example (Figure 3.2), the computation of L_2 is the equivalent of performing Pythagoras theorem on *xdist* and *ydist* and yields the "crow flies" distance between the points. Although this metric captures our intuitions about space and is widely used (Kaufman & Rousseeuw, 1990;

Cutting et al., 1992; Schütze, 1993; Schroeder & Noy, 2001), it has been shown to be less effective than the L_1 Norm. The reason for this, discussed by Lee (1999), is that it is quadratic in the verbs not shared by the two nouns and, as noted by Kaufman and Rousseeuw (1990) it is "extremely sensitive to the effect of one or more outliers". In other words, a large difference in one dimension will have a strong influence on the overall score since the differences in each dimension are squared. Further increases in n lead to further increases of this effect, until L_∞ is reached, which is defined as the maximal distance between the points in any single dimension:

$$dist_{L_\infty}(n_1, n_2) = \max_v |P(v|n_1) - P(v|n_2)| \tag{3.6}$$

3.1.1.2 The Cosine Measure

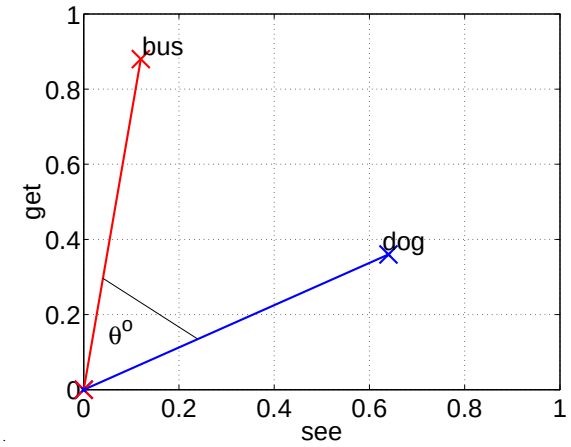

Figure 3.3: The cosine measure

The cosine measure returns the cosine of the angle between the vectors (see Figure 3.3) and is calculated as the dot product of the vectors:

$$cos(n_1, n_2) = \frac{\sum_v P(v|n_1).P(v|n_2)}{\sqrt{\sum_v P(v|n_1)^2 \sum_v P(v|n_2)^2}} \tag{3.7}$$

The angle between vectors representing identical distributions is zero and so the maximum value of the cosine measure is 1. A cosine value of zero indicates complete lack of similarity since vectors of words with no shared co-occurrences will be orthogonal.

This measure is very popular as a vector similarity measure (Salton & McGill, 1983; Schütze, 1998; Manning & Schütze, 1999; Caraballo, 1999; Padó & Lapata, 2003; Wu & Zhou, 2003), but was shown by Lee (1999) to be less effective than other measures as a lexical distributional similarity measure since it incorporates information about non-shared co-occurrences.

3.1.1.3 Other Vector Based Techniques

In our discussion so far, we have not considered the number of dimensions in our semantic or context space. Following (Lin, 1998a; Lee, 1999), we consider every verb co-occurrence with our target nouns and therefore every verb (in the corpus) is a possible dimension. It is possible to reduce the dimensionality by considering, say, only the n most frequent co-occurrence types as dimensions (Padó & Lapata, 2003) or by using a mathematical technique such as singular value decomposition (SVD) (Deerwester et al., 1990; Schütze, 1992b, 1992a, 1998). In SVD (see Forsyte,

Malcolm, and Moler (1977) for an introduction), analysis begins with a matrix of associations between pairs of objects. For example, pairs of objects might be noun-verb co-occurrences and association values might be their observed frequencies or probabilities. SVD is a general matrix analysis technique, similar to eigen-analysis for symmetric matrices, which finds the independent components or principal axes of variation in the data. The minor axes of variation can be ignored, leading to a reduced dimensionality.

The use of such a technique leads to efficiency improvements that may be particularly important when implementing computationally expensive clustering algorithms. However, in this work we focus on the comparison of different similarity measures, and in general, with modern hardware, the computational issues of retaining the full dimensionality are less significant. Another potential advantage of using SVD is that by finding the major axes of variation, it is possible to abstract away from noise and detect underlying features. However, we expect that a good similarity measure should perform well relative to other similarity measures independent of any common preprocessing of the data. Further, the use of SVD might increase the apparent performance of similarity measures that would otherwise focus on insignificant features or noise. Hence, in order to compare similarity measures, we use the full dimensionality of the original data. Later (Section 3.1.5), we will consider the use of mutual information which can be viewed as a way of detecting the most significant features.

3.1.2 Correlation Coefficients

Two random variables are said to be positively correlated if a high value in one would lead us to expect a high value in the other, and, conversely, a low value in one would lead us to expect a low value in the other. In the context of lexical distributional similarity, correlation has been used in particular by Schütze (1998). We can envisage plotting a scatter graph where each point represents a possible verb co-occurrence and its position relative to the x-axis represents its probability or frequency of co-occurrence with one noun, and its position relative to the y-axis represents its probability or frequency of co-occurrence with another noun. For similar nouns (e.g. *bus* and *train*) we would expect to find positive correlation (Figure 3.4) whereas for dissimilar nouns (e.g. *bus* and *dog*) we would expect to find zero (or negative) correlation (Figure 3.5).

3.1.2.1 Pearson's Product-Moment Correlation Coefficient

The most well known correlation coefficient is Pearson's product-moment correlation coefficient. It measures the extent to which the values of two variables lie on the same side of their respective means. If we use the variable X to represent the probability distribution $P(v|n_1)$ and the variable Y to represent the probability distribution $P(v|n_2)$, then the correlation or similarity between the two nouns can be computed using the standard definition for Pearson's product-moment correlation coefficient:

$$sim_{pm} = \frac{\sum(X-\overline{X})(Y-\overline{Y})}{\sqrt{\sum(X-\overline{X})^2\sum(Y-\overline{Y})^2}} \tag{3.8}$$

where $\overline{X}$ is the arithmetic mean of the X distribution and $\overline{Y}$ is the arithmetic mean of the Y distribution. The coefficient lies in the range [-1,1] where -1 represents perfect negative correlation and 1 represents perfect positive correlation. This measure is closely related to the cosine measure (Sec-

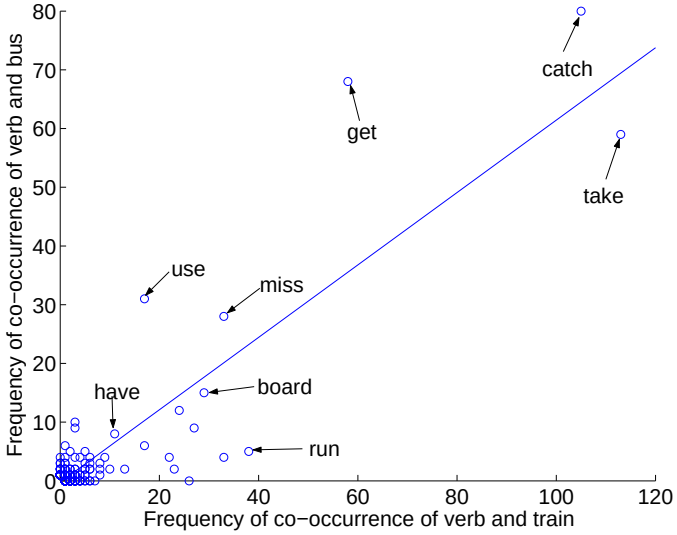

Figure 3.4: Positive correlation between bus and train

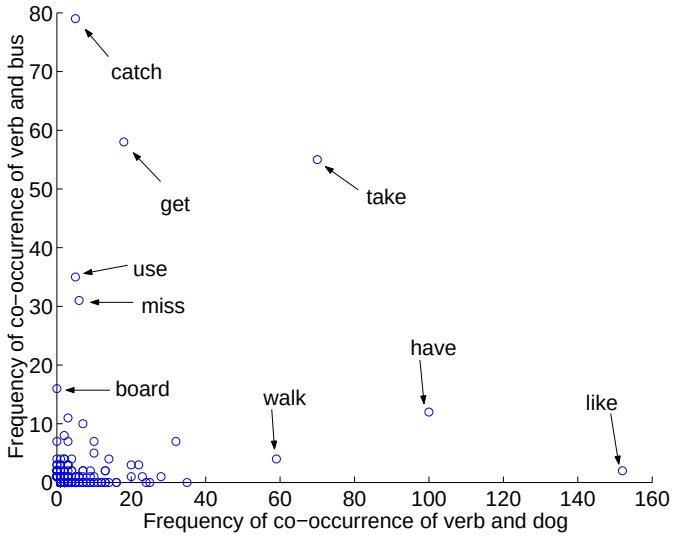

Figure 3.5: Correlation between bus and dog

tion 3.1.1.2) since we are effectively finding the cosine between two vectors whose values have been normalised by subtracting the respective means of the original values. Further, we would expect the means of our conditional probability distributions (or co-occurrence frequency distributions) to be very close to zero since there will always be a large number of verbs which do not occur at all with a given noun. In this setting, Pearson's product-moment correlation coefficient is approximated by the cosine measure. This also explains why we do not expect to see (much) negative correlation between words since we do not expect to see a verb co-occurring with a noun less than the mean number of times (zero).

3.1.2.2 Rank Correlation Coefficients

A potential problem with using Pearson's product-moment correlation coefficient in this area is that it assumes a parametric distribution, i.e. one distributed normally around a mean. As discussed by Pedersen (1996), this assumption of normality is shared by many common statistical tests and

is rarely valid in NLP where distributions tend to contain large numbers of rare events. Such is the case for lexical co-occurrence distributions (see Baayen (2001) for a thorough introduction to the nature of word frequency distributions). One non-parametric solution is to use a correlation coefficient based on rankings such as Spearman's rank correlation coefficient or Kendall's τ coefficient. For example, a noun might rank verbs according to their co-occurrence frequency. Any verbs that co-occur the same number of times with the noun are given the arithmetic mean of the ranks they jointly occupy.

Spearman's rank correlation coefficient, used by Finch and Chater (1992) to find syntactic categories, is simply defined as the product-moment correlation coefficient between the two sets of rankings. However, since the values being correlated are ranks, the formula can usually be simplified to:

$$sim_{spearman} = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{3.9}$$

where d_i is the difference in ranks for the ith item and n is the number of items. However, this formula is only an approximation when there are tied ranks. Further, using the formula or the definition, we are likely to run into more problems because of the large number of verbs that are tied at the bottom of the ordering. These verbs will all occur below the mean of the ranking and therefore verbs that occur with neither noun will have a strong influence on the apparent positive correlation between the nouns.

Kendall's τ coefficient (Gibbons, 1993; Hatzivassiloglou & McKeown, 1993; Lee, 1997) also just considers the ordering put on the co-occurrence types by the nouns. This measure tests whether noun n_1 occurring more frequently with verb v_1 than verb v_2 is correlated with noun n_2 occurring more frequently with v_1 than v_2. More formally, for every pair of co-occurrence types (v_i, v_j), we consider the signs of the quantities $P(v_i|n_1) - P(v_j|n_1)$ and $P(v_i|n_2) - P(v_j|n_2)$. If both signs are the same, then the verb pair is a *concordance*; if the signs are different, the pair is a *discordance*; and if either quantity is zero, then the pair is a *tie*. sim_τ is the difference between the probability of observing a concordance and the probability of observing a discordance. It can be estimated using:

$$sim_\tau = \frac{\text{number of observed concordances} - \text{number of observed discordances}}{|V|(|V|-1)} \tag{3.10}$$

where V is the set of all co-occurrence types. This coefficient has the advantage over Spearman's rank correlation coefficient in that ties only serve to reduce the value of the correlation towards zero. However, it is very expensive to compute. For every pair of nouns, every pair of verbs must be considered. This contrasts to the majority of measures, which can be computed, for each pair of nouns, in linear time with the number of verbs.

3.1.3 Combinatorial Measures

Combinatorial similarity measures are based on counting shared and non-shared features. The nature of these measures is such that, in their simplest forms at least, the extent to which each object has each feature is not considered[1]. In the context of lexical distributional similarity, we

[1] It is possible however, as we will discuss later, to incorporate weight functions into these measures.

can consider a word to be a feature of another word if the two have co-occurred together in some grammatical relation. Since we consider just noun-verb direct-object co-occurrences, the features of noun n are verbs, v, and are given by $F(n)$:

$$F(n) = \{v : P(v|n) > 0\} \tag{3.11}$$

To compute the similarity of two nouns n_1 and n_2, we can consider the relative sizes of the regions in Figure 3.6. The sizes of the four regions are given by the cardinality of the intersections specified in Table 3.2. Different measures combine these values in different ways. We will now discuss some of the possibilities.

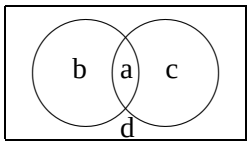

Figure 3.6: Venn Diagram for similarity between n_1 and n_2

	$F(n_1)$	$\overline{F(n_1)}$
$F(n_2)$	a	b
$\overline{F(n_2)}$	c	d

Table 3.2: Sizes of intersections of feature sets

3.1.3.1 Jaccard's Coefficient

Jaccard's coefficient (Salton & McGill, 1983; Lee, 1999; Curran & Moens, 2002a, 2002b), also known as the Tanimoto coefficient (Bensch & Savitch, 1992; Resnik, 1993), can be defined as the proportion of features belonging to either noun that are shared by both nouns, that is, the ratio between the size of the intersection of the feature sets and the size of the union of feature sets:

$$sim_{jacc}(A,B) = \frac{|F(A) \cap F(B)|}{|F(A) \cup F(B)|} = \frac{a}{a+b+c} \tag{3.12}$$

3.1.3.2 The Dice Coefficient

The Dice Coefficient (Frakes & Baeza-Yates, 1992; Kay & Röscheisen, 1993; Smadja, McKeown, & Hatzivassiloglou, 1996; Curran & Moens, 2002a) is similar to Jaccard's Coefficient, differing only in that it gives twice the weight to the shared features. It computes the ratio between the size of the intersection of the two feature sets and the sum of the sizes of the individual feature sets:

$$sim_{dice}(A,B) = \frac{2.|F(A) \cap F(B)|}{|F(A)| + |F(B)|} = \frac{2a}{2a+b+c} \tag{3.13}$$

It was shown by van Rijsbergen (1979) that Dice and Jaccard's coefficient are monotonic in each other. This means that although the scores computed by each will be different, the orderings or rankings put on objects will be the same. Hence, in many applications, there is little to be gained by using one of these measures over the other.

3.1.3.3 Other Combinatorial Measures

There are a large number of other measures that can be computed from the values in Table 3.2, some of which are reviewed in Ravichandran and Rao (2001). Many of these incorporate the number of features that neither object has (d). For example, there is the simple *matching* coefficient, which can be computed as:

$$sim_{matching} = \frac{a+d}{a+b+c+d} \tag{3.14}$$

and *Russell and Rao's* coefficient (Ravichandran & Rao, 2001), computed as:

$$sim_{rr} = \frac{a}{a+b+c+d} \tag{3.15}$$

However, in the context of lexical distributional similarity, we would expect the number d to be much larger than the other values. Hence, $sim_{matching}$ will tend towards 1 for all words and sim_{rr} will tend towards 0 for all words.

3.1.4 Substitutability Measures

Similarity between two nouns, n_1 and n_2, can be considered as a measure of the appropriateness of substituting n_1 for n_2. Since the appropriateness of substituting n_2 for n_1 may be different to the appropriateness of substituting n_1 for n_2, this leads to a number of asymmetric measures which we now consider. We also consider the symmetric Jensen-Shannon divergence measure, which is based on the asymmetric Kullback-Leibler divergence measure. Throughout, when referring to asymmetric measures, we shall use the notation n_2 to denote the target noun and n_1 to denote a potential neighbour of n_2.

3.1.4.1 Kullback-Leibler Divergence Measure

The Kullback-Leibler divergence measure or relative entropy (Kullback & Leibler, 1951; Cover & Thomas, 1991; Pereira et al., 1993; Dagan et al., 1999; Clark, 2000) can be used to calculate the distance between two probability mass functions. The relative entropy, $(D(p||q))$, between two distributions p and q is defined (Cover & Thomas, 1991) as "the inefficiency of assuming that the distribution is q when the true distribution is p." Inefficiency is defined in information-theoretic terms and hence, relative entropy is measured in bits:

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{3.16}$$

where we assume the mathematical equivalences $0 \log \frac{0}{q} = 0$ and $p \log \frac{p}{0} = \infty$.

It can be shown (Cover & Thomas, 1991) that relative entropy is always non-negative and is equal to zero if and only if the two distributions are identical, making it a potentially useful distance measure. If, as in previous examples, we are measuring the similarity between nouns n_1 and n_2 based on their co-occurrences with verbs in the direct-object relation, we can set $p = P(v|n_2)$ and $q = P(v|n_1)$ and measure the similarity between nouns n_1 and n_2 as the inverse of the distance between distributions p and q, i.e.:

$$dist_{kl}(n_2, n_1) = D(P(v|n_2)||P(v|n_1) = \sum_v P(v|n_2) \log \frac{P(v|n_2)}{P(v|n_1)} \tag{3.17}$$

Informally, this measure is the average log ratio of the corresponding probabilities in each distribution, weighted by their probabilities in the true distribution. It is obvious from the mathematical definition that relative entropy is asymmetric i.e. $D(p||q) \neq D(q||p)$. $dist_{kl}(n_2, n_1)$ measures the cost of using n_1 instead of n_2 whilst $dist_{kl}(n_1, n_2)$ measures the cost of using n_2 instead of n_1. Hence, we use $dist_{kl}(n_2, n_1)$ to measure the fitness of n_1 as a substitute for n_2 and $dist_{kl}(n_1, n_2)$ to measure the fitness of n_2 as a substitute for n_1. In other words, to find the neighbours of n_2 we find the nouns n_1 which minimise $D(P(v|n_2)||P(v|n_1))$. The intuition as to why the distance in one direction might be less than in the other is discussed by Lee (1997); we would expect a flat or uniform distribution to be a better model for a sharp empirical distribution than a sharp model distribution is for a flat empirical distribution.

A second issue, which must be considered when measuring the similarity between two nouns using relative entropy, is that $D(P(v|n_2)||P(v|n_1)) = \infty$ if there are any verbs v such that $P(v|n_2) > 0$ and $P(v|n_1) = 0$. Accordingly, we cannot use relative entropy directly on maximum likelihood estimate (MLE) probabilities (unless we wish to find that most nouns are infinitely different to each other). We must either smooth the base language model probability estimates or use a related measure which can be used directly on the MLE probabilities. We will now discuss two such measures: the Jensen-Shannon divergence measure and the α-skew divergence measure.

3.1.4.2 Jensen-Shannon Divergence Measure

The Jensen-Shannon divergence measure (Rao, 1983; Lin, 1991) is a popular alternative to the Kullback-Leibler Divergence measure which has frequently been used in word clustering and nearest neighbour techniques (Dagan et al., 1999; Lapata et al., 2001; Dhillon et al., 2002; Schulte Im Walde & Brew, 2002). It can be defined (Dagan et al., 1999) as "the average of the KL divergence of each of the two distributions to their average distribution":

$$dist_{js}(p,q) = \frac{1}{2}\left[D\left(p||\frac{p+q}{2}\right) + D\left(q||\frac{p+q}{2}\right)\right] \tag{3.18}$$

Like the Kullback-Leibler divergence measure on which it is based, the Jensen-Shannon divergence measure is always non-negative and equal to zero only when the two distributions are identical. However, in contrast to the Kullback-Leibler divergence measure, it is a symmetric measure since it measures the cost of using the average distribution in place of each of the individual distributions. This has two clear advantages over relative entropy. First, it does not require smoothed probability estimates since the mean of two probability values is greater than zero unless both values are zero. Second, it can be shown (Dagan et al., 1999) that it only depends on the verbs which occur with both nouns. Whilst the computational advantage of this is obvious, Lee (1999) showed that measures which depend only on the contexts which occur with both target words perform better in her pseudo-disambiguation experiments.

3.1.4.3 α-Skew Divergence Measure

The α-skew divergence measure (Lee, 1999, 2001) is a popular approximation to the Kullback-Leibler divergence measure, which, like the Jensen-Shannon Divergence Measure, is frequently used as a measure of lexical distributional similarity (Even-Zohar & Roth, 2000; McCarthy, 2000; Schulte Im Walde & Brew, 2002; Padó & Lapata, 2003). It is an approximation developed to be used when unreliable MLE probabilities would result in the actual Kullback-Leibler divergence

measure being equal to ∞. It is defined (Lee, 1999) as:

$$dist_{\alpha}(q,r) = D(r||\alpha.q + (1-\alpha).r) \tag{3.19}$$

for $0 \leq \alpha \leq 1$. In effect, the q distribution is smoothed with the r distribution, which results in it always being non-zero when the r distribution is non-zero. The parameter α controls the extent to which the measure approximates the Kullback-Leibler divergence measure. When α is close to 1, the approximation is close whilst avoiding the problem with zero probabilities associated with using the Kullback-Leibler divergence measure. This theoretical justification for using a very high value of α (e.g. 0.99) is also borne out by empirical evidence (Lee, 2001).

However, unlike the Jensen-Shannon divergence measure, the α-skew divergence measure retains the asymmetry of the Kullback-Leibler divergence measure. Accordingly, it is necessary to consider in which direction it should be computed. In defence of the asymmetry of the α-skew divergence measure, Lee (1999) comments:

> But the substitutability of one word for another need not be symmetric. For instance, "fruit" may be the best possible approximation to "apple", but the distribution of "apple" may not be a suitable proxy for the distribution of "fruit".

The Kullback-Leibler divergence measure $D(p||q)$ measures the cost of using distribution q instead of distribution p. Since it is effectively calculated over the verbs occurring in the p distribution (since the term for any verb not occurring in the p distribution will be zero), it is the second argument, i.e. the model distribution q, which needs to be smoothed. In other words, to find the neighbours of the noun n_2 we would expect to find those nouns n_1 which minimise $dist_{\alpha}(P(v|n_1),P(v|n_2))$.

However, Lee (2001) obtains better results on a pseudo-disambiguation task by minimising $dist_{\alpha}(P(v|n_2),P(v|n_1))$. Lee justifies using the measure in this direction, saying that, since it is the distribution of the target word, n_2 that we will be using the neighbours to smooth, it is this distribution which is unreliable and requires smoothing. However, this no longer fits with the notion of measuring how well the proxy distribution approximates the actual distribution.

Thus, we have a conflict. Minimising $dist_{\alpha}(P(v|n_1),P(v|n_2))$ retains the meaning of the KL divergence whilst minimising $dist_{\alpha}(P(v|n_2),P(v|n_1))$ would appear to produce the best results. We shall return to this later when we make our empirical comparison of the different similarity measures (Section 3.2). For the time being, we shall consider the α-skew divergence measure to be two separate measures and compute two sets of neighbours for each word accordingly. We will find the neighbours of n_2 by computing:

$$dist_{\alpha 1}(n_1,n_2) = dist_{\alpha}(P(v|n_1),P(v|n_2)) \tag{3.20}$$

and:

$$dist_{\alpha 2}(n_1,n_2) = dist_{\alpha}(P(v|n_2),P(v|n_1)) \tag{3.21}$$

Intuitively, the difference between using $dist_{\alpha 1}$ and $dist_{\alpha 2}$ can be viewed as follows. Low values of $dist_{\alpha 1}$ arise when the probabilities of the contexts given the potential neighbour are similar to the probabilities of the contexts given the target word for contexts in which the *target*

word occurs frequently. Low values of $dist_{\alpha 2}$, on the other hand, arise when the probabilities of the contexts given the potential neighbour are similar to the probabilities of the contexts given the target word for contexts in which the *potential neighbour occurs frequently*.

3.1.4.4 Confusion Probability

The confusion probability (Sugawara, Nishimura, Toshioka, Okachi, & Kaneko, 1985; Essen & Steinbiss, 1992; Grishman & Sterling, 1993; Dagan et al., 1999; Lapata et al., 2001) is an estimate of the probability that one word can be substituted for another. The intuition is that nouns n_1 and n_2 are completely confusable if we are equally as likely to see n_1 occurring as the object of a given verb token as we are to see n_2 occurring as the object of that verb token. In other words, knowledge of the verb does not change the relative probabilities of n_1 and n_2. If we wish to compute the probability that word n_1 can be substituted for n_2 based on verb direct-object co-occurrences, then we could use:

$$sim_{cp}(n_1|n_2) = \sum_v \frac{P(n_2|v).P(n_1|v).P(v)}{P(n_2)} \tag{3.22}$$

As discussed by Dagan et al. (1999), this expression can be rewritten using Bayes' rule and computed as:

$$sim_{cp}(n_1|n_2) = \sum_v \frac{P(v|n_2)}{P(v)}.P(v|n_1).P(n_1) \tag{3.23}$$

Since it is a probability, it lies in the range [0,1] where 1 represents maximal confusability or similarity, and 0 represents complete lack of confusability or similarity. Like the Kullback-Leibler divergence measure, it is asymmetric. However, it has the curious property that a word is not necessarily the most similar word to itself. This is because it is high when the potential neighbour occurs frequently with verbs which occur more often with the target word than one would expect. Accordingly, it will tend to be higher for *potential neighbours* with high unigram probabilities.

In our empirical comparison, we will also consider what we will refer to as the reverse confusion probability:

$$sim_{cp_{rev}}(n_1|n_2) = \sum_v \frac{P(v|n_1)}{P(v)}.P(v|n_2).P(n_2) \tag{3.24}$$

In contrast to sim_{cp}, we expect this to be high when the potential neighbour occurs more often than one would expect with verbs which occur frequently with the target word. Accordingly, it will be higher for *target nouns* with high unigram probabilities but this will not change the rankings of the potential neighbours of a given target noun.

3.1.5 Mutual Information Based Measures

Mutual information allows us to capture the intuition that some verb co-occurrences are more informative than others. For example, some verbs, such as *get* and *have*, are selectionally weak in the constraints they place on their arguments (Resnik, 1993) which means that we would expect to see them co-occurring with nearly every noun. Therefore, the fact that they have co-occurred with both nouns under consideration may tell us less about the similarity of those two nouns than if those two nouns both occur as the object of a selectionally strong verb such as *eat*.

More formally, mutual information is (Cover & Thomas, 1991):

> a measure of the amount of information one random variable contains about another random variable. It is the reduction in uncertainty of one random variable due to the knowledge of the other.

In our example, seeing the verb *eat* reduces our uncertainty about what the object of the verb will be more than seeing the verb *have*, i.e. it is more informative about its object.

The *pointwise mutual information* between two words can be used to measure their relatedness or degree of association (Church & Hanks, 1989). For a noun n and verb v it can be written as:

$$I(v,n) = \log \frac{P(v,n)}{P(v).P(n)} = \log \frac{P(v|n)}{P(v)} \tag{3.25}$$

Intuitively, it is a measure of how much the probability that the verb is v is increased by knowing that the noun is n (or, since it is symmetric, how much the probability that the noun is n is increased by knowing that the verb is v). Negative values indicate that the probability that the verb is v decreases if we know that the noun is n and a value of zero indicates that the verb and the noun occur together no more or less frequently than one would expect by chance.

We now consider particular measures of similarity that use mutual information.

3.1.5.1 Hindle's Measure

Hindle (1990) proposed an MI based measure which he used to show that nouns could be reliably clustered based on their verb co-occurrences. If we are considering just noun-verb direct-object co-occurrences, the measure can be simplified to:

$$sim_{hind}(n_1,n_2) = \sum_v f(I(v,n_1),I(v,n_2)) \tag{3.26}$$

where:

$$f(a,b) = \begin{cases} min(a,b) & \text{if } a > 0 \text{ and } b > 0 \\ abs(max(a,b)) & \text{if } a < 0 \text{ and } b < 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.27}$$

Intuitively, the more highly informative contexts shared by the two nouns, the higher their similarity. However, similarity is also increased between the two nouns if there are verbs with which they both co-occur less than would be expected. Theoretically, this causes a problem with verbs which have not occurred with the noun at all. These verbs have also occurred less with the noun than would be expected by chance but cannot contribute to the similarity since the MI between events that do not occur together is undefined or equal to $-\infty$. Accordingly, Hindle's measure can only be calculated over the verbs which occur with both nouns. As a consequence of this, we adopt the variant proposed by Lin (1998a) which only considers the verbs which have positive MI with both nouns:

$$sim_{hind+}(n_1,n_2) = \sum_{v \in T(n_1) \cap T(n_2)} min(I(v,n_1),I(v,n_2)) \text{ where } T(n) = \{v : I(v,n) > 0\} \tag{3.28}$$

3.1.5.2 Lin's MI-based Measure

Lin (1998a) proposed a measure of lexical distributional similarity based on his information-theoretic similarity theorem (Lin, 1997, 1998c):

> The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are.

Since we are considering just noun-verb direct-object co-occurrence data, the similarity between two nouns n_1 and n_2 can be calculated using Lin's measure as:

$$sim_{lin}(n_1,n_2) = \frac{\sum_{T(n_1)\cap T(n_2)}(I(n_1,v)+I(n_2,v))}{\sum_{T(n_1)} I(n_1,v)+\sum_{T(n_2)} I(n_2,v)} \text{ where } T(n) = \{v : I(n,v) > 0\} \tag{3.29}$$

Using this measure (over a greater number of grammatical relations), Lin (1998a) showed that thesaurus entries could be generated which were more akin to entries derived from WordNet than if entries were derived using Roget, sim_{hind+} or variants of the cosine measure, Jaccard's coefficient or the Dice coefficient. More recently, Wiebe (2000) has used the measure to successfully identify subjective adjectives.

3.1.5.3 Other MI Based Approaches

Mutual information is a popular and intuitive measure of lexical association which has led to its incorporation in variations of a number of standard similarity measures. For example, Lin (1998a) defines the features of a noun to be those contexts for which the MI between the noun and the context are positive and then uses the combinatorial Dice and Jaccard's coefficients on those feature sets. Further, a generalisation of Jaccard's coefficient (Grefenstette, 1994b) allows a weight to be incorporated on each feature. We can also envisage measuring a cosine between vectors made up of (positive) MI values rather than conditional probabilities. Curran and Moens (2002a) explore a number of weights, including MI, for the cosine measure, Jaccard's Coefficient and the Dice Coefficient.

However, as discussed by Resnik (1993), Fung and McKeown (1997),Kilgarriff and Tugwell (2001b), Weeds and Weir (2003a) and Wu and Zhou (2003), MI tends to over-emphasise the association of low frequency events. Wu and Zhou (2003) proposes a variation Lin's similarity measure which replaces the MI scores with weighted mutual information (WMI) (Fung & McKeown, 1997) scores:

$$\text{WMI}(v,n) = P(v,n)\log\frac{P(v,n)}{P(v).P(n)} = P(v,n)\log\frac{P(v|n)}{P(v)} \tag{3.30}$$

Since the pointwise MI is multiplied by the probability of the co-occurrence to give the WMI score, the association of low frequency co-occurrences will be reduced more than the association of high frequency co-occurrences.

3.2 Empirical Comparison of Measures

In this section, we will empirically compare fifteen of the measures introduced in the previous section. Table 3.3 lists the measures and the notation used to refer to them. We will first discuss

the data used to derive the neighbour sets. We will then look at similarity in the rankings of neighbours by different neighbours across 2000 nouns. Finally, we will look at the tendencies of certain measures to select high or low frequency neighbours and discuss the impact this may have on applications.

notation	description
$dist_{L_1}$	the L_1 Norm (Section 3.1.1.1)
$dist_{L_2}$	the L_2 Norm or Euclidean distance (Section 3.1.1.1)
sim_{cos}	the cosine measure (Section 3.1.1.2)
sim_{pm}	Pearson's product-moment correlation coefficient (Section 3.1.2.1)
sim_{τ}	Kendall's τ correlation coefficient (Section 3.1.2.2)
sim_{jacc}	Jaccard's coefficient (Section 3.1.3.1)
$sim_{jacc_{mi}}$	Jaccard's coefficient using MI to select features (Section 3.1.5.3)
sim_{dice}	the Dice coefficient (Section 3.1.3.2)
$dist_{js}$	the Jensen-Shannon divergence measure (Section 3.1.4.2)
$dist_{\alpha 1}$	Lee's α-skew divergence measure (Section 3.1.4.3)
$dist_{\alpha 2}$	Lee's α-skew divergence measure computed in the opposite direction (Section 3.1.4.3)
sim_{cp}	the confusion probability (Section 3.1.4.4)
$sim_{cp_{rev}}$	the confusion probability computed in the opposite direction (Section 3.1.4.4)
sim_{hind+}	Hindle's similarity measure using only positive MI (Section 3.1.5.1)
sim_{lin}	Lin's similarity measure (Section 3.1.5.2)

Table 3.3: Fifteen similarity measures

3.2.1 Data

We first describe the data used and then explain the rationale behind its use.

The data from which sets of nearest neighbours are derived is direct-object data for 2000 nouns extracted from the British National Corpus (BNC) using a robust accurate statistical parser (RASP) (Briscoe & Carroll, 1995; Carroll & Briscoe, 1996). Briscoe and Carroll (2002) gives a full description of the system. In summary, RASP is a pipelined modular system containing modules for: tokenisation, part-of-speech (PoS) and punctuation tagging, lemmatisation, parsing and statistical disambiguation. Of particular interest to us is the parsing module, which parses a lattice of PoS and punctuation tags using a manually developed wide-coverage grammar. The output of the system can be displayed as a set of (ranked) trees and/or as a sequence of grammatical relations between lexical heads. It is the latter that we use as raw data for our research. The RASP system is designed to be robust, since it parses part-of-speech and punctuation tags rather than word tokens and the accuracy has been shown (Briscoe & Carroll, 2002) to be competitive with that of highly lexicalised parse selection models. Table 3.4 (Carroll, 2003) summarises the reliability with which different relations are extracted by RASP.

The original data set extracted by the parser consisted of 2,852,300 possibly indistinct lemmatised (noun,verb) direct-object pairs. From this we selected data pertaining to a set of 2000 nouns (1,995,998 pairs in total). We set aside 20% of the data for each noun as test data for future evaluation tasks. The remaining 80% (consisting of 1,596,798 co-occurrence tokens distributed over

331,079 distinct co-occurrence types) was used to calculate similarities between all pairs of nouns according to each of the measures listed in Table 3.3. We now discuss why we use direct-object data (Section 3.2.1.1) and then why and how we selected the set of 2000 nouns (Section 3.2.1.2).

3.2.1.1 Direct Object Data

In this work we look at similarity between nouns. The use of grammatical relations to model context precludes finding similarities between words of different parts of speech. Since we are looking at similarity in terms of substitutability, we would not expect to find a word of one part of speech substitutable for a word of another part of speech. However, we could have easily chosen to look at similarity between verbs or between adjectives using the same techniques. We chose to look at nouns on the basis that one cannot do everything at once and one has to start somewhere. Further, nouns tend to allow less sense extensions than verbs and adjectives (Pustejovsky, 1995), and the noun hyponymy hierarchy in WordNet, which will be used as a pseudo-gold standard for comparison, has been extensively used in previous research (Lin, 1998a; Clark & Weir, 2002; Moldovan, Harabagiu, Peters, Guthrie, & Wilks, 2001).

Some previous work on distributional similarity between nouns has used only a single grammatical relation, usually the direct-object relation (Lee, 1999), whereas other work has considered multiple grammatical relations (Lin, 1998a). An argument for using multiple grammatical relations is that, otherwise, we may miss similarities between certain pairs of nouns. It may be the case that two nouns do not occur as the object of any of the same verbs but they do occur as the subject of some of the same verbs.

We use just direct-object data for two reasons. The first is that we are using automatically extracted data. Using grammatical relations, such as the indirect object, which are unreliably extracted by the parser, would introduce more noise into our data.

The second reason is that it is not clear that including other relations will improve the neighbours obtained and, if other relations are to be included, how the information should be combined. Further, since we have no notion of words being semantically similar in different roles, it would make little sense if we could make such a distinction in terms of distributional similarity. The only difference we would expect to find is one of granularity.

For example, the only classification we might expect to obtain from the distribution of determiners with respect to nouns is that of countability (Baldwin & Bond, 2003a), since determiners only select for this feature. Further, we might expect verbs to be more selective in their object position than in their subject position. Since a large number of verbs take only human, or possibly animate, subjects, it could be argued that the main use of subject data would be to distinguish animate and non-animate concepts — something which we would expect object data to be able to do anyway. In order to investigate whether including more data from the non-clausal subject relation in our descriptions of nouns would be advantageous, we conducted a preliminary experiment as follows:

We selected direct-object noun-verb pairs and non-clausal subject[2] noun-verb pairs from the data described above for the 1000 most frequent nouns (frequency calculated over the subject and object position in the data-set and considering all nouns, not just those occurring in WordNet). We then constructed three data sets as follows:

[2]Both relations are extracted with approximately equal accuracy by RASP (Table 3.4).

Relation	Precision	Recall	F-score	# test GRs
dependent	75.82	76.81	76.31	6624.00
mod	76.96	73.90	75.40	3759.00
ncmod	76.55	69.63	72.92	2213.00
xmod	75.00	44.53	55.88	76.00
cmod	60.37	47.37	53.08	164.00
detmod	92.82	91.99	92.40	1114.00
arg_mod	0.00	0.00	0.00	0.00
arg	72.31	82.61	77.12	2326.00
subj	82.79	84.53	83.65	1069.00
ncsubj	84.11	84.68	84.40	1045.00
xsubj	100.00	80.00	88.89	4.00
csubj	16.67	33.33	22.22	6.00
subj_or_dobj	83.87	83.93	83.90	1457.00
comp	63.40	80.59	70.97	1257.00
obj	60.34	78.67	68.30	764.00
dobj	**86.86**	**82.40**	**84.57**	**388.00**
obj2	29.63	84.21	43.84	54.00
iobj	33.64	68.35	45.09	321.00
clausal	68.15	83.37	75.00	493.00
xcomp	83.64	84.16	83.90	324.00
ccomp	69.89	80.25	74.71	93.00
aux	89.04	87.40	88.21	374.00
conj	69.09	69.09	69.09	165.00

Table 3.4: Accuracy of RASP

dobj 1,948,198 direct-object relation noun-verb pairs

ncsubj 2,240,701 non-clausal subject relation noun-verb pairs

sep 4,188,899 direct-object and non-clausal subject noun-verb pairs mixed together where co-occurrence with the same verb in subject and object position generates a count of two separate features

Each data set was split 80:20 between training and test data. For each training set, the similarity between each pair of nouns was calculated using $dist_{\alpha 2}$ with $\alpha = 0.99$. We used this measure since it had been demonstrated to be effective for the 1000 most frequent nouns using a pseudo-disambiguation task (Lee, 2001). Our pseudo-disambiguation experiment[3] was set up as follows.

Separate direct-object *dobj* and non-clausal subject *ncsubj* test sets were constructed using the paradigm described in Chapter 2. We discarded all noun-verb pairs from the test data that were also seen in the training data. The resulting *dobj* and *ncsubj* test sets contained 23,238 pairs and 33,614 pairs respectively[4]. Each remaining noun-verb pair (n, v_1) was replaced with a noun-verb-

[3]Pseudo-disambiguation experiments will be discussed in more detail in Chapter 5.

[4]This represents a reduction in test set size, due to discarding *seen* pairs, of over 90%.

verb triple (n, v_1, v_2) where v_2 is selected such that it is of approximately equal frequency to v_1 in the training data and does not occur with n in the training data. The task is then for the k-nearest neighbours of n to decide which of v_1 and v_2 was the original co-occurrence. In order to do this we use a frequency-weighted voting scheme. Each neighbour of n is given a vote which it casts to the verb with which it co-occurs most frequently and which is weighted by the difference in frequency with which it co-occurs with each verb. Votes are totted up over all of the k nearest neighbours of n where k is a parameter to be optimised. Performance is measured as error rate, defined as:

$$\text{Error} = \frac{\text{\# of incorrect choices} + \frac{\text{\# of ties}}{2}}{\text{\# of test instances}} \tag{3.31}$$

where a tie results when the neighbours cannot decide between the two alternatives.

The performance of nearest neighbours derived from *dobj*, *ncsubj* and *sep* evaluated over the test sets *dobj* and *ncsubj* is shown in Figure 3.7. For each combination of neighbour set and test set, we show the variation in error rate with the neighbourhood size, k. We would expect better neighbour sets to outperform poorer neighbour sets at all values of k, and, in particular, at low values of k.

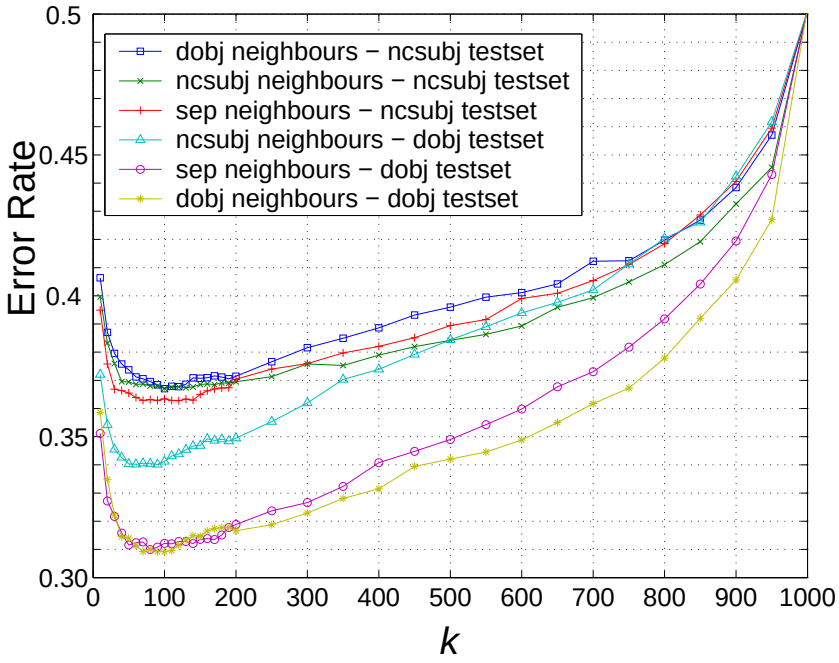

Figure 3.7: Performance of Neighbour Sets on Different Test Sets

Based on Figure 3.7, we make the following three observations:

1. All three types of neighbour sets do better at disambiguating the direct object test instances than the non-clausal subject test instances. This supports the hypothesis that verbs are more selective in their object position than in their subject position. Nearest neighbours, of any sort, are less able to disambiguate the non-clausal subject verbs because they tend to either co-occur with both verbs or with neither verb.

2. The direct object derived neighbours perform as well as the non-clausal subject derived neighbours on the *ncsubj* test set and better on the *dobj* test set. This supports the hypothesis that nouns that are similar in their object co-occurrences are also similar in their subject co-occurrences.

3. Increasing the amount of training data by including information from both relations did not lead to significantly improved results over using information solely from the best-performing relation. This suggests that the non-clausal subject data is not adding any new information about the nouns, i.e., the difference in distributional similarity according to these two grammatical relations is one of granularity. Hence, we decided to perform the rest of our experiments using just the direct-object relation data.

3.2.1.2 2000 Nouns

We will first explain how we selected the 2000 nouns and then explain our rationale.

The complete set of 2000 nouns is made up of a subset of 1000 high frequency and a subset of 1000 low frequency nouns. We computed the frequency of nouns by counting the number of times they occurred in the direct-object relation data. We used counts based only on this data since some nouns occur more frequently in one relation than another, and therefore frequency of occurrence in the direct-object data reflects most accurately the amount of data we have for each noun. Having constructed a frequency list, we discarded all nouns not also occurring in WordNet1.6 (since this will be used for some of our evaluation experiments). We then selected the 1000 most frequent nouns as the high frequency subset. The frequencies of these nouns lie in the range [586,20871]. For the low frequency nouns, we selected the nouns with frequency ranks 3001–4000. This corresponds to a frequency range of [72,121]. See Appendix A for a full listing of these nouns.

We will now consider why we study these nouns.

Much previous work in this area considers either a small selection of nouns or only the most frequent nouns. For example, Lee (1999) and Lee (2001) considers the 1000 most frequently occurring nouns. Curran and Moens (2002a) considers 70 nouns which are randomly chosen from WordNet such that they cover a range of values for certain word properties (including frequency). Lin (1998a) is perhaps the most extensive study, considering similarities between 4294 nouns. However, these nouns were still selected as the most frequent nouns (occurring more than 100 times in Lin's parsed corpus), and there is no analysis of performance for words of different frequencies within that set of nouns.

However, as noted by Rosenfeld (2000), it is low frequency nouns that stand to gain the most by distributional clustering techniques, since these are the words for which one would most likely want to overcome the sparse data problem. A distributional clustering technique which only works well for high frequency nouns may be less useful than one which works well for low frequency nouns. We study a set of high frequency nouns and a set of low frequency nouns in order to investigate whether the techniques under consideration perform significantly differently according to the relative frequency of the word.

As will be discussed in much more detail in Section 3.2.3, there is, however, a potential problem with only considering a subset of nouns (although it may be argued that whatever set we use, it will only ever be a subset of the universe of nouns). The problem is that the actual, or *true*, neighbours of a word might not fall within that set. In other words, a neighbour set generated from a more complete set of, say, 30,000 nouns might look very different from the neighbour set generated by considering similarities between 2000 nouns. There are a number of reasons for restricting the number of nouns considered as potential neighbours. The first is purely computational — it is much easier to compute 2000×2000 similarities than 2000×5000 or $2000 \times 30,000$ similarities.

Second, by not considering the mid-frequency nouns, we make a clear differentiation between high and low frequency nouns (rather than saying a word of frequency n is low frequency but a word of frequency $n+1$ is high frequency).

Having described the construction of and the rationale behind our data set, we will now discuss the method used to compare distributional similarity measures.

3.2.2 Cross-Comparison of Neighbours Derived Using Different Measures

For every possible noun-noun combination in the set of 2000 nouns and using the data described in Section 3.2.1.1, we computed the similarity between the two nouns using each of the measures in Table 3.3. We sorted the potential neighbours for each noun and measure by similarity in order to create ranked neighbour sets. Since it is pointless in the majority of applications to consider a word as a potential neighbour of itself, we preclude that possibility in our neighbour sets. However, if given the chance, every measure except the two confusion probability measures would make every word the nearest neighbour of itself. Table 3.5 shows the ten nearest neighbours of the word *hope* according to each of the measures.

Although it is dangerous to make general observations based on a single word, we can see that the product-moment correlation coefficient duplicates the neighbours produced by the cosine measure (both coloured red) and the Dice Coefficient duplicates the neighbours produced by Jaccard's coefficient (both coloured green)[5]. Apart from these duplicates, the huge variety in neighbour sets possible is evident. From 13 distinct measures there are a total of 63 different words suggested as being in the top ten neighbours of hope. Further, there are only four words (highlighted in bold in Table 3.5) that are suggested by more than four (distinct) measures: **feeling** $\times 9$, **dream** $\times 7$, **confidence** $\times 7$ and **desire** $\times 6$.

In order to make a more general and quantitative comparison of sets of neighbours produced by different measures, we use the following technique which is adapted from Lin (1998a).

3.2.2.1 Neighbour Set Comparison Technique

In order to compare two neighbour sets of size k, we transform each neighbour set so that each neighbour is given a rank score of $k - rank$. Unlike Lin (1998a), we do not use the similarity scores directly since these require normalization if scores computed on different scales are to be compared. Having performed this transformation, which could be considered a normalization, two neighbour sets for the same word w may be represented by two ordered sets of words $[w_k, ..., w_1]$ and $[w'_k, ..., w'_1]$. The similarity between such sets is computed using the same calculation as used by Lin (1998a) except for simplifications due to the use of ranks:

$$C([w_k, ..., w_1], [w'_k, ..., w'_1]) = \frac{\sum_{w_i = w'_j} i \times j}{\sum_{i=1}^{k} i^2} \tag{3.32}$$

where i and j are the rank scores of the words within each neighbour set. Using this comparison technique, we are in fact computing the cosine between neighbour set vectors. If we are considering N words as potential neighbours, then there are N dimensions in our neighbour set space. Potential neighbours not within a given rank distance k of the noun score zero, otherwise the score is $k - rank$.

[5] The relevance of the other colour groupings will become clear in the next section

$dist_{L_1}$	$dist_{L_2}$	sim_{cos}	sim_{pm}	sim_{τ}
confidence 1.14	value 0.14	flavour 0.87	flavour 0.87	idea 0.028
dream 1.15	**feeling 0.14**	**feeling 0.87**	**feeling 0.87**	time 0.028
feeling 1.16	child 0.14	reason 0.87	reason 0.87	life 0.028
desire 1.16	meaning 0.15	wish 0.86	wish 0.86	interest 0.028
reason 1.17	thought 0.15	plenty 0.86	plenty 0.86	work 0.028
vision 1.17	**confidence 0.15**	meaning 0.86	meaning 0.86	people 0.027
baby 1.18	cause 0.15	misgiving 0.86	misgiving 0.86	power 0.027
opportunity 1.18	**desire 0.15**	**desire 0.85**	**desire 0.85**	man 0.027
doubt 1.18	authority 0.15	sympathy 0.85	sympathy 0.85	thing 0.027
flavour 1.18	memory 0.15	value 0.85	value 0.85	part 0.027

$dist_{js}$	$dist_{\alpha 1}$	$dist_{\alpha 2}$	sim_{cp}	$sim_{cp_{rev}}$
reason 0.30	value 1.70	inkling 1.46	view 0.010	hem 0.013
feeling 0.23	**feeling 1.71**	reason 1.46	hand 0.010	dissatisfaction 0.009
vision 0.31	idea 1.71	plenty 1.50	concern 0.009	dismay 0.009
idea 0.31	view 1.80	qualm 1.59	idea 0.009	scepticism 0.008
doubt 0.31	vision 1.82	regard 1.61	time 0.009	concern 0.007
confidence 0.31	interest 1.85	overtone 1.65	interest 0.009	outrage 0.007
plenty 0.31	fear 1.87	chance 1.67	way 0.008	beak 0.006
chance 0.32	**dream 1.88**	misgiving 1.67	effect 0.008	warrior 0.006
dream 0.32	thought 1.88	laugh 1.70	money 0.007	optimism 0.006
faith 0.32	power 1.89	doubt 1.73	**feeling 0.007**	readiness 0.006

sim_{jacc}	sim_{dice}	$sim_{jacc_{mi}}$	sim_{hind}	sim_{lin}
dream 0.25	**dream 0.41**	**dream 0.17**	**dream 91**	**dream 0.29**
chance 0.25	chance 0.40	ambition 0.16	spirit 64	ambition 0.25
confidence 0.24	**confidence 0.39**	**confidence 0.16**	**desire 62**	**confidence 0.23**
commitment 0.24	commitment 0.39	love 0.14	**feeling 60**	love 0.21
love 0.24	love 0.38	expectation 0.14	opposition 60	sense 0.20
concern 0.23	concern 0.38	opposition 0.13	ambition 60	**desire 0.20**
ambition 0.22	ambition 0.37	**desire 0.13**	interest 60	expectation 0.19
sense 0.22	sense 0.37	pride 0.13	fear 59	doubt 0.19
feeling 0.22	**feeling 0.36**	concern 0.13	**confidence 59**	suspicion 0.19
expectation 0.22	expectation 0.36	fear 0.13	memory 57	**feeling 0.18**

Table 3.5: Neighbours of *hope*, with their associated similarity scores, according to each similarity measure

Given our earlier comments on cosines and correlation coefficients, when $k = N$, this is also similar to computing Spearman's rank order correlation coefficient between the neighbour sets. However, we are unlikely to want to calculate the correlation over all potential neighbours since, generally, we are not concerned about the order in which a measure ranks dissimilar words. Using a lower value of k, we are still approximating subtracting the means and computing the product-moment correlation coefficient between the rank scores but, since a large number of items have

score zero, we will not find negative correlation between any sets. In other words, the correlation lies between 0 and 1, where 1 represents maximum similarity between the sets. To achieve 1, exactly the same words must appear in the two sets in exactly the same order.

For every pair of measures, we calculated the similarity between neighbour sets of size 200 for every noun in our set of 2000 nouns. We then computed the mean and standard deviation for the high frequency nouns and for the low frequency nouns. These results for each are in Tables 3.6 and 3.7 respectively. For compactness, each measure is represented by just the subscript given in Table 3.3. We also do not include the product-moment correlation coefficient and the Dice coefficient in this table since there was 100% correlation between each of these measures and the cosine measure and Jaccard's coefficient respectively for all 2000 nouns.

If neighbour sets were created randomly, we would expect to see an average correlation (over 1000 nouns) of 0.08 with standard deviation 0.02. Accordingly, for the majority of measures, the overlap seen between neighbour sets is significantly greater than one would expect by chance[6]. The most significant overlap is seen in three groups of measure. The first is the MI-Based measures: $sim_{jacc_{mi}}$, sim_{hind} and sim_{lin}. The second group is made up of $dist_{L_1}$, sim_{js}, $dist_{\alpha 1}$ and $dist_{\alpha 2}$. All of the measures in this second group consider the difference in the extent to which each noun co-occurs with each verb. Finally, there is considerable overlap between sim_{τ} and sim_{cp}. These groups are coloured magenta, cyan and blue respectively in Tables 3.5, 3.6 and 3.7.

However, although the overlap between different measure's neighbour sets is, in general, significant, it is perhaps the variation which is more significant. In the next section we will look at one of the ways in which neighbours selected by different measures vary — that is in their frequency relative to the target noun.

3.2.3 High or Low Frequency Neighbours?

In this section, we look at the tendencies of certain measures to select high(er) or low(er) frequency words as neighbours of the target noun. In order to do this, we compare the neighbour sets produced when all 2000 nouns are considered as potential neighbours with when only the 1000 high frequency nouns are considered as potential neighbours and with when only the 1000 low frequency nouns are considered as potential neighbours.

There are other possible ways of analysing the frequency characteristics of neighbour sets. For example, we could calculate the average frequency of the top k neighbours or calculate the proportion of the top k neighbours that are high frequency or low frequency. The neighbour set comparison technique, which we use, is appealing for a number of reasons. First, it is consistent with our work on comparing different similarity measures. Second, it weights the importance of neighbours by their rank in the set, so a large number of high frequency neighbours in the positions closest to the target noun is considered more biased than a large number of high frequency neighbours distributed throughout the neighbour set. Third, the maximum score, 1, tells us that the two neighbour sets are identical. Thus, we can easily generalise the results about frequency characteristics to application performance. If the neighbour sets derived from the 1000 most frequent nouns are identical to those derived from the set of "all" nouns, we know that in future we

[6] Statistical significance at the 95% level is achieved at around 0.09 although the exact figure depends on the measure due to the variation in standard deviation observed.

	L_2	*cos*	τ	*js*	α1	α2	*cp*	cp_{rev}	*jacc*	$jacc_{mi}$	*hind*	*lin*
L_1	0.64 (0.11)	0.65 (0.14)	0.43 (0.13)	0.93 (0.03)	0.80 (0.09)	0.74 (0.10)	0.49 (0.08)	0.56 (0.15)	0.39 (0.15)	0.61 (0.17)	0.60 (0.18)	0.58 (0.16)
L_2		0.56 (0.19)	0.34 (0.10)	0.58 (0.11)	0.57 (0.10)	0.39 (0.15)	0.31 (0.07)	0.26 (0.16)	0.32 (0.12)	0.37 (0.14)	0.43 (0.15)	0.34 (0.14)
cos			0.24 (0.08)	0.64 (0.12)	0.51 (0.14)	0.65 (0.10)	0.34 (0.08)	0.40 (0.20)	0.30 (0.12)	0.33 (0.14)	0.33 (0.15)	0.34 (0.15)
τ				0.49 (0.12)	0.65 (0.13)	0.32 (0.10)	0.76 (0.07)	0.12 (0.05)	0.49 (0.24)	0.35 (0.14)	0.52 (0.09)	0.36 (0.12)
js					0.87 (0.05)	0.82 (0.08)	0.56 (0.07)	0.40 (0.15)	0.61 (0.14)	0.63 (0.15)	0.63 (0.16)	0.62 (0.15)
α1						0.62 (0.11)	0.66 (0.05)	0.34 (0.14)	0.63 (0.16)	0.60 (0.15)	0.67 (0.14)	0.59 (0.15)
α2							0.45 (0.08)	0.43 (0.15)	0.50 (0.12)	0.53 (0.14)	0.47 (0.15)	0.55 (0.14)
cp								0.29 (0.09)	0.44 (0.18)	0.35 0.13	0.48 (0.10)	0.40 (0.11)
cp_{rev}									0.25 (0.09)	0.33 (0.10)	0.32 (0.10)	0.43 (0.12)
jacc										0.76 (0.08)	0.68 (0.11)	0.69 (0.10)
$jacc_{mi}$											0.80 (0.11)	0.88 (0.05)
hind												0.81 (0.11)

Table 3.6: Mean cross measure correlation of neighbour sets for high frequency nouns. Figures in brackets are standard deviations

need only consider the 1000 most frequent nouns as potential neighbours. We will now explain the technique used in more detail.

	L_2	*cos*	τ	*js*	α1	α2	*cp*	cp_{rev}	*jacc*	$jacc_{mi}$	*hind*	*lin*
L_1	0.46 (0.15)	0.69 (0.14)	0.23 (0.09)	0.84 (0.06)	0.58 (0.11)	0.59 (0.15)	0.30 (0.09)	0.43 (0.16)	0.43 (0.13)	0.46 (0.13)	0.51 (0.17)	0.50 (0.16)
L_2		0.56 (0.19)	0.34 (0.10)	0.47 (0.15)	0.57 (0.11)	0.21 (0.17)	0.32 (0.07)	0.23 (0.16)	0.08 (0.06)	0.10 (0.07)	0.36 (0.12)	0.18 (0.09)
cos			0.23 (0.09)	0.74 (0.09)	0.54 (0.15)	0.60 (0.13)	0.32 (0.09)	0.38 (0.20)	0.21 (0.11)	0.23 (0.14)	0.36 (0.16)	0.30 (0.14)
τ				0.33 (0.11)	0.58 (0.14)	0.15 (0.06)	0.70 (0.08)	0.13 (0.06)	0.04 (0.04)	0.04 (0.05)	0.38 (0.10)	0.16 (0.10)
js					0.74 (0.09)	0.70 (0.12)	0.41 (0.08)	0.43 (0.16)	0.35 (0.12)	0.35 (0.12)	0.55 (0.17)	0.48 (0.14)
α1						0.37 (0.15)	0.55 (0.07)	0.34 (0.17)	0.17 (0.10)	0.17 (0.12)	0.60 (0.13)	0.36 (0.14)
α2							0.26 (0.08)	0.37 (0.15)	0.36 (0.11)	0.35 (0.11)	0.31 (0.13)	0.40 (0.12)
cp								0.28 (0.09)	0.04 (0.05)	0.04 (0.05)	0.38 (0.09)	0.173 (0.08)
cp_{rev}									0.36 (0.10)	0.39 (0.11)	0.49 (0.11)	0.55 (0.11)
jacc										0.85 (0.05)	0.38 (0.10)	0.69 (0.08)
$jacc_{mi}$											0.41 (0.10)	0.74 (0.08)
hind												0.67 (0.10)

Table 3.7: Mean cross measure correlation of neighbour sets for low frequency nouns. Figures in brackets are standard deviations

For each noun *n* and similarity measure, *sim* (where *sim* may be one of the distance measures), two new sets of ranked neighbours were constructed: one set where neighbour nouns are restricted

to being in the high frequency subset of nouns ($m_{sim}^{high}(n)$) and the other set where neighbour nouns are restricted to being in the low frequency subset of nouns ($m_{sim}^{low}(n)$). Each of the new neighbour sets, like the complete neighbour set ($m_{sim}^{comp}(n)$) is of size $k = 200$. We then used the neighbour set comparison technique (Section 3.2.2.1) to compare the similarity between neighbour sets for a single measure. Since the two new sets are mutually exclusive, their similarity must be zero. However, by comparing the complete neighbour set ($m_{sim}^{comp}(n)$) where neighbours are of any frequency to the restricted frequency neighbour sets, we get an estimate of how the complete neighbour set is made up of high and low frequency nouns.

If the complete neighbour set is made up completely of high frequency nouns then the correlation with the high frequency neighbour set is 1 (since the neighbours must be in the same order) and the correlation with the low frequency neighbour set is 0. Conversely, if the complete neighbour set is made up completely of low frequency nouns then the correlation with the high frequency neighbour set is 0 and the correlation with the low frequency neighbour set is 1. If there are no biases towards high or low frequency nouns, i.e. there is an even proportion of high and low frequency neighbours in the complete neighbour set, we would expect to get an average correlation of 0.624 (standard deviation 0.042) with both high and low frequency neighbour sets. The reason why the correlation is greater than 0.5 is that the shared neighbours will be at the top of the restricted frequency neighbour sets i.e. neighbours at the bottom of the restricted frequency neighbour sets will not be in the complete neighbour set unless the complete neighbour set is identical to that restricted frequency neighbour set. Due to this, we calculate a normalised score, which is the bias towards high frequency neighbours for a noun, n, by a measure, sim:

$$biashigh_{sim}(n) = \frac{C(m_{sim}^{comp}(n), m_{sim}^{high}(n))}{C(m_{sim}^{comp}(n), m_{sim}^{high}(n)) + C(m_{sim}^{comp}(n), m_{sim}^{low}(n))} \tag{3.33}$$

The value lies in the range [0,1] where 1 indicates a neighbour set completely made up of high frequency nouns, 0 indicates a neighbour set completely made up of low frequency nouns and 0.5 indicates a neighbour set with no biases towards high or low frequency nouns.

We compute this normalised score for every noun (and measure) and consider averages for high and low frequency nouns so that we can see whether the neighbours of high frequency nouns tend to be high or low frequency nouns and whether the neighbours of low frequency nouns tend to be high or low frequency nouns. The tendencies of each measure to select high frequency neighbours for high and low frequency nouns are given in Table 3.8.

As can be seen, all of the measures, with the (possible) exceptions of $dist_{L_1}$,$sim_{cp_{rev}}$ and $dist_{\alpha 2}$, have strong tendencies to select neighbours of particular frequencies. We can categorize the measures by whether, in relation to the frequency of the target noun, they tend to select high(er) frequency neighbours for all nouns, low(er) frequency neighbours for all noun or neighbours of similar frequencies for all nouns. These tendencies are summarized in Table 3.9.

These tendencies may not be as surprising as they might seem at first sight if one considers the relationship between word frequency and distribution width, that is the number of distinct contexts a word occurs in. A word cannot have been observed in more distinct contexts than it has been observed in indistinct contexts, i.e., $width < frequency$.

But, although frequency puts an upper bound on width, one might expect to find very narrow words occurring very frequently. This however is rarely the case, as is shown by the 4×4 con-

	Frequency of target noun *n*	
	high	low
$dist_{L_1}$	0.916	0.556
$dist_{L_2}$	0.903	0.867
sim_{cos}	0.903	0.866
sim_{τ}	1.000	0.999
$dist_{js}$	0.935	0.695
$dist_{\alpha 1}$	0.980	0.903
$dist_{\alpha 2}$	0.809	0.519
sim_{cp}	0.998	0.990
$sim_{cp_{rev}}$	0.437	0.348
sim_{jacc}	0.990	0.208
$sim_{jacc_{mi}}$	0.946	0.139
sim_{hind}	0.988	0.679
sim_{lin}	0.853	0.380

Table 3.8: Average value of $biashigh_{sim}$ for high frequency target nouns and for low frequency target nouns. Values above 0.5 indicate that the measure is biased towards selecting high frequency nouns as neighbours (for those target nouns) and values less than 0.5 indicate that the measure is biased towards selecting low frequency nouns as neighbours (for those target nouns).

high(er) frequency selecting measures	low(er) frequency selecting measures	similar frequency selecting measures
sim_{τ}	$sim_{cp_{rev}}$	sim_{jacc}
sim_{cp}		$sim_{jacc_{mi}}$
$dist_{L_2}$		sim_{lin}
$dist_{\alpha 1}$		
sim_{hind}		
$dist_{js}$		
sim_{cos}		
? $dist_{L_1}$		
? $dist_{\alpha 2}$		

Table 3.9: Measures grouped by tendency to select high frequency, low frequency or similar frequency neighbours

tingency table in 3.10. This contingency table was constructed for the 1000 most frequent nouns since these are the nouns for which we could expect highest independence between width and frequency (they all have a frequency in excess of 576).

Under the null hypothesis of uniformity (i.e. independence between frequency and width), we would expect to find 62.5 nouns in each of the sixteen groups. Performing a χ^2 test on this table shows, however, that the observed correlation between frequency and width is extremely significant ($\chi^2_{calc} = 903.9$, $\chi^2_{5\%}(9) = 23.519$).

Thus, the most frequent nouns are also the widest nouns and the least frequent nouns are

	Frequency				
Width	high	high-med	med-low	low	total
high	195	55	0	0	250
high-med	43	136	61	10	250
med-low	8	43	121	78	250
low	4	16	68	162	250
total	250	250	250	250	1000

Table 3.10: Contingency table showing relationship between distribution width and frequency for 1000 most frequent nouns

also the narrowest nouns. Accordingly, the similar frequency selecting measures are selecting words that have occurred in a similar number of distinct contexts; the higher frequency selecting measures are selecting words that have occurred in a greater number of distinct contexts; and the lower frequency selecting measures are selecting words that have occurred in a smaller number of distinct contexts. From this it would appear that different distributional similarity measures are implicitly tied to different plausible hypotheses about the nature of the nearest neighbours of a word. Higher frequency selecting measures work on the assumption that the best substitute for a noun *n* is one which can be used in all of the contexts of that noun *n* and it does not matter if they can be used in other contexts. Lower frequency selecting measures work on the assumption that the best substitute for a noun *n* is one which can only be used in the contexts of that noun but it does not matter if it cannot be used in all of the contexts. Similar frequency selecting measures attempt to make a trade-off between the two extreme positions. It is one of the goals of the rest of this thesis to discover which position is best in the context of particular applications. Our hypotheses, see Chapter 1, would suggest that higher frequency selecting measures will be better at semantic tasks and lower frequency selecting measures will be better at distributional tasks.

Returning to the existing measures under consideration, some of their frequency characteristics might be predicted from their definitions. For example, all of the "similar frequency" selecting measures consider a ratio between what the nouns share and the entire descriptions of the two nouns. The only way to maximise such a ratio is to minimise the features not shared by the nouns. In other words, the nouns must have a similar number of features. The other measures tend to focus more on the commonality of the two nouns or on the commonality of the two nouns in relation to one noun's features. This allows more scope for a difference in the number of features each noun has.

However, whatever the actual cause of the tendencies observed, they still have the same effect. For the vast majority of the measures, there is little to be gained by considering all words as potential neighbours of a word. As part of our preliminary experiments, we performed the pseudo-disambiguation task described in 3.2.1.1 for a random set of 1000 words (frequency lying in the range [2, 20561]). However, we used four different similarity measures, $dist_{\alpha 1}$, $dist_{\alpha 2}$, $dist_{L_1}$ and sim_{lin}, and for each measure and each noun we calculated three different neighbour sets. One neighbour set was restricted to neighbours from the set of *most frequent* nouns, one was restricted to neighbours from the *random* set of nouns and the third was calculated from all of the nouns appearing at least twice in our direct-object relation data extracted from the BNC. There are a

total of 45,181 nouns in this data and neighbours taken from this set can be considered to be the target noun's *true* neighbours with respect to the similarity measure used. We performed the pseudo-disambiguation task for a sequence of values of k (the neighbourhood size) up to $k = 999$. The results (minimum error rates and corresponding values of k) for each measure and set of neighbours are summarized in Table 3.11.

	$dist_{\alpha 1}$	$dist_{\alpha 2}$	$dist_{L_1}$	sim_{lin}
true neighbours	0.290 (k=68)	0.339 (k=999)	0.300 (k=60)	0.266 (k=167)
random neighbours	0.368 (k=8)	0.386 (k=50)	0.377 (k=30)	0.345 (k=36)
most frequent neighbours	0.290 (k=40)	0.309 (k=30)	0.298 (k=30)	0.270 (k=50)

Table 3.11: Error rates on pseudo-disambiguation task for four measures using *true* neighbours and two restricted neighbour sets. Figures in brackets are the values of k, the neighbourhood size, at which the minimum error rates were found.

The increase in error rate observed when neighbours are no longer taken from the entire set of nouns but instead from a set of 1000 random nouns, is what one might intuitively expect: to get the best results, one has to consider every noun as a potential neighbour. However, when neighbours are restricted to being from the *most frequent* noun set, the error rate is much lower than for the random noun set. This means that the *most frequent* nouns can in no way be considered a random set of nouns. Only for one of the measures, sim_{lin}, is there any increase in performance in using the *true* neighbours over the *most frequent* neighbours. For two of the measures ($dist_{\alpha 2}$ and $dist_{L_1}$), there is a decrease in performance.

Performance for $dist_{\alpha 1}$ and $dist_{\alpha 2}$ is particularly interesting. Earlier in this chapter, we commented that $dist_{\alpha 1}$ retains the meaning of the KL divergence measure whilst $dist_{\alpha 2}$ has been shown (Lee, 2001) to obtain the best results of the two at pseudo-disambiguation. This, however, is not what can be observed here. However, recall that we consider a set of 1000 random target nouns. Lee's results, on the other hand, are for neighbours of the 1000 most frequent nouns (calculated from amongst the 1000 most frequent nouns). Accordingly, to investigate this, we calculated neighbours for the 1000 most frequent nouns from amongst the 1000 most frequent nouns and from amongst all nouns using $dist_{\alpha 1}$ and $dist_{\alpha 2}$. Our results are in Table 3.12.

	$dist_{\alpha 1}$	$dist_{\alpha 2}$
true neighbours	0.327 (k=80)	0.328 (k=500)
most frequent neighbours	0.328 (k=80)	0.314 (k=120)

Table 3.12: Error rates on pseudo-disambiguation task for $dist_{\alpha 1}$ and $dist_{\alpha 2}$ measures using *true* neighbours and *most frequent* neighbours. Figures in brackets are the values of k at which the minimum error rates were found.

We have now managed to achieve the same result as Lee (2001). However, it is clear from the rest of the results in this section that $dist_{\alpha 2}$ only performs well when all of the words, i.e. target words and potential neighbours, are high frequency words. Accordingly, in the majority of our experiments we will concentrate on results for $dist_{\alpha 1}$ as opposed to $dist_{\alpha 2}$.

3.3 Conclusions

In this chapter, we have discussed a broad range of popular measures of distributional similarity. We have compared the neighbours that a number of them produce from the same data and examined their tendencies to select high or low frequency words as neighbours. We have seen that Jaccard's Coefficient and the Dice Coefficient produce identical neighbour sets as do the cosine metric and the product-moment correlation coefficient, but otherwise there is large variety in the neighbour sets produced by different measures. The majority of measures tend to select words as nearest neighbours which are of a higher frequency or have a greater distribution width than the target word. This fits with the intuition that a good substitute for a word should be able to occur in at least all of the contexts in which the target word occurs.

Knowledge of the tendencies of measures to select words of particular frequencies as neighbours might also be used to explain or predict their behaviour in application tasks. This knowledge may have computational benefits. For example, only consider the highest frequency words as potential neighbours if a measure is only going to select neighbours from amongst that set. Further, some applications might benefit from using a distributional similarity measure with particular frequency characteristics. For example, a distributional similarity measure which tends to select higher frequency neighbours would be much more appropriate in a text simplification task (see Chapter 2). However, frequency characteristics may also have a negative impact, particularly if we wish to compare different types of distribution. For example, a proposed word sense separation algorithm might compare the similarity of neighbour descriptions to the target noun's sense descriptions. However, we expect a neighbour's distribution to be at least as wide as the target noun's distribution (using 14 out of the 15 measures). Hence, it will be wider than all of the target noun's sense descriptions and there will be a bias towards finding every neighbour to be related to the widest, most frequent sense of the target noun.

In the next chapter we will present a new general framework for distributional similarity in which we can explore systematically the questions we have about distributional similarity. First, is substitutability, and therefore similarity, asymmetric? Second, are some contexts more important than others, and, if so, which ones?

Chapter 4

A General Framework for Distributional Similarity

In this chapter, we present a new general framework for distributional similarity. This framework directly defines a similarity function, does not require smoothing of the base language model and allows us to explore systematically the questions about similarity raised in Chapter 2. In this framework, similarity between words is viewed as a measure of how appropriate it is to use one word (or its distribution) in place of the other. Like relative entropy, it is inherently asymmetric since we can measure how appropriate it is to use word A instead of word B separately from how appropriate it is to use word B instead of word A.

In Section 4.2, we illustrate the framework with neighbours selected for the noun *hope* and examine the tendencies of different models within the framework to select high or low frequency words as neighbours. In the third section, we will describe the characteristics of existing distributional similarity measures from Chapter 3 with reference to the framework.

4.1 Co-occurrence Retrieval

Let us imagine that we have formed descriptions of each noun in terms of the verbs they co-occur with in the direct object relation in some corpus. Now let us imagine that we have lost (or accidentally deleted) the description for noun n_2 but before this happened we had noticed that the description of noun n_2 was very similar to that of noun n_1. Hence, we decide that we can use the description of noun n_1 instead of the description of noun n_2 and hopefully nobody will notice. How well we do will depend on the validity of substituting n_1 for n_2, or, in other words, the similarity between n_1 and n_2.

The task we have set ourselves can be seen as *co-occurrence retrieval* (CR). By analogy with information retrieval, where there is a set of documents that we would like to retrieve and a set of documents that we do retrieve, we have a scenario where there is a set of co-occurrences that we would like to retrieve, the co-occurrences of n_2, and a set of co-occurrences that we have retrieved, the co-occurrences of n_1. Continuing the analogy, we can measure how well we have done in terms of precision and recall, where precision tells us how much of what was retrieved was correct and recall tells us how much of what we wanted to retrieve was retrieved.

Our general framework for distributional similarity is based on this concept of co-occurrence

retrieval. As the distribution of word B moves away from being identical to that of word A, it can be *similar* in one of two ways: it can have occurred in generally more distinct contexts than word A; or, it can have occurred in generally less distinct contexts than word A. In the first case, we call word B a high recall neighbour of word A; and, in the second case, we call word B a high precision neighbour of word A. We can explore the merits of symmetry and asymmetry in a similarity measure by varying the relative importance attached to precision and recall.

The remainder of this section is devoted to defining six co-occurrence retrieval models (CRMs). These models can be divided into two types. The first is the simpler and is based on the Boolean concept of two objects either sharing or not sharing a particular feature (where objects are nouns and features are verb co-occurrences). This type of model will be referred to as an *additive model*. The second type of model incorporates the difference in extent to which each noun has each feature, and this type of model will be referred to as a *difference-weighted model*. Exploring the two types of models, both defined on the same concepts of precision and recall, allows us to investigate the question posed in Chapter 2, is a shared context worth the same regardless of the difference in the extent to which each word appears in that context?

We also use the CR framework to investigate the second question posed in Chapter 2, "should all contexts be treated equally?" by using different potential word association[1] functions within each type of model. We use association functions to decide which verbs are features of a noun and also to determine the relative importance of features. We use combinatorial, probabilistic and MI based association functions which allow us to define type-based, token-based and MI-based CRMs respectively. Using these models we can explore giving equal weight to each verb co-occurrence, more weight to more frequently occurring verb co-occurrences and more weight to more informative verb co-occurrences.

4.1.1 Additive Models

We have considered the intuition behind calculating precision and recall for co-occurrence retrieval, we now formulate this more formally in terms of an additive model.

We first need to consider for each noun n which verb co-occurrences will be retrieved, or predicted, by it and, conversely, required in a description of it. We will refer to these verbs as the features of n, $F(n)$:

$$F(n) = \{v : D(n,v) > 0\} \tag{4.1}$$

where $D(n,v)$ is the degree of association between noun n and verb v. Possible association functions will be described in the context of particular CRMs.

The shared features of noun n_1 and noun n_2 are referred to as the set of True Positives (TP):

$$TP = F(n_1) \cap F(n_2) \tag{4.2}$$

The precision of n_1's retrieval of n_2's features is the proportion of n_1's features that are shared by both nouns, where each feature is weighted by its relative importance according to n_1:

[1] We use the term *association* to describe the degree or extent to which two words co-occur together.

$$\mathcal{P}^{add}(n_1, n_2) = \frac{\sum_{TP} D(n_1, v)}{\sum_{F(n_1)} D(n_1, v)} \tag{4.3}$$

The recall of n_1's retrieval of n_2's features is the proportion of n_2's features that are shared by both nouns, where each feature is weighted by its relative importance according to n_2:

$$\mathcal{R}^{add}(n_1, n_2) = \frac{\sum_{TP} D(n_2, v)}{\sum_{F(n_2)} D(n_2, v)} \tag{4.4}$$

Precision and recall both lie in the range [0,1] and are both equal to one when each noun has exactly the same features. It should also be noted that the recall of n_1's retrieval of n_2 is equal to the precision of n_2's retrieval of n_1, i.e., $\mathcal{R}^{add}(n_1, n_2) = \mathcal{P}^{add}(n_2, n_1)$.

The role of the association function should be considered carefully. It first decides which co-occurrences of n_1 and n_2 are important enough to put in a description of it, or by analogy with document retrieval, which co-occurrences we want to retrieve and which co-occurrences we have retrieved. It is then used to weight contexts by their importance. In the latter case, $D(n_1, v)$ tells us the retrieval process's perceived relevance of verb v and $D(n_2, v)$ tells us the actual relevance of verb v.

We will now define three possible additive models for co-occurrence retrieval: a type-based CRM, a token-based CRM and a mutual information based CRM. Mathematically, the differences between these models are the association functions used to determine and weight features.

4.1.1.1 Additive Type-Based CRM

In this model, the precision of n_1's retrieval of n_2 is the proportion of distinct verbs (or verb co-occurrence types) which occur with n_1 that also occur with n_2; and the recall of n_1's retrieval of n_2 is the proportion of distinct verbs (or verb co-occurrence types) which occur with n_2 that also occur with n_1. Alternatively, we can say that the precision of n_1's retrieval of n_2 is the probability that a verb type picked at random from those co-occurring with n_1 also occurs with n_2; and the recall of n_1's retrieval of n_2 is the probability that a verb type picked at random from those co-occurring with n_2 also occurs with n_1.

We formalise the (additive) type-based CRM using a combinatorial association function. The degree of association (D_{type}) between a noun n and a verb v is 1 if they have co-occurred together and 0 otherwise:

$$D_{type}(n, v) = \begin{cases} 1 & \text{if } P(v|n) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

In this case, it should be noted that the value of D for each feature in each sum in the definitions of precision and recall must be 1. Hence, the definitions can be simplified as follows:

$$\mathcal{P}^{add}_{type}(n_1, n_2) = \frac{\sum_{TP} D_{type}(n_1, v)}{\sum_{F(n_1)} D_{type}(n_1, v)} = \frac{|TP|}{|F(n_1)|} \tag{4.6}$$

$$\mathcal{R}^{add}_{type}(n_1, n_2) = \frac{\sum_{TP} D_{type}(n_2, v)}{\sum_{F(n_2)} D_{type}(n_2, v)} = \frac{|TP|}{|F(n_2)|} \tag{4.7}$$

These simplified expressions correspond with the intuitions presented at the start of this section: precision is the proportion of co-occurrence types which occur with n_1 that also occur with

n_2; and recall is the proportion of co-occurrence types which occur with n_2 that also occur with n_1.

4.1.1.2 Additive Token-Based CRM

In this model, in contrast to the type-based model, we are interested in retrieving (possibly) indistinct verbs, i.e., verb co-occurrence tokens. The precision of n_1's retrieval of n_2 is the proportion of indistinct verbs (or verb co-occurrence tokens) which occur with n_1 that also occur with n_2; and the recall of n_1's retrieval of n_2 is the proportion of indistinct verbs (or verb co-occurrence tokens) which occur with n_2 that also occur with n_1. Alternatively, we can say that the precision of n_1's retrieval of n_2 is the probability that a verb token picked at random from those co-occurring with n_1 also occurs with n_2; and the recall of n_1's retrieval of n_2 is the probability that a verb token picked at random from those co-occurring with n_2 also occurs with n_1.

We can formalise this model using a probabilistic association function. The degree of association (D_{tok}) between a noun n and verb v is defined in the token-based CRM as:

$$D_{tok}(n,v) = P(v|n) \tag{4.8}$$

Hence, nouns have the same features as in the type-based CRM but each feature is given a weight based on its probability of occurrence. Alternatively, we could use a frequency based association function $D_{tok} = f(v,n)$, where $f(v,n)$ is the frequency with which verb v and noun n have co-occurred. However, this leads to the same expressions for precision and recall since:

$$P(v|n) = \frac{f(v,n)}{f(n)} \tag{4.9}$$

We use the probabilistic association function since it highlights the fact that the values of precision and recall are independent of the (absolute) amount of data considered for each noun. Further, using the probabilistic association function allows us to easily see that the definitions for precision and recall in this model can be simplified. Basic probability theory tells us that:

$$\sum_v P(v|n) = 1 \tag{4.10}$$

and since $F(n) = \{v : D(n,v) > 0\} = \{v : P(v|n) > 0\}$, it follows that:

$$\sum_{F(n)} D_{tok}(n,v) = 1 \tag{4.11}$$

and therefore:

$$\mathcal{P}_{tok}^{add}(n_1,n_2) = \frac{\sum_{TP} D_{tok}(n_1,v)}{\sum_{F(n_1)} D_{tok}(n_1,v)} = \sum_{TP} P(v,n_1) \tag{4.12}$$

$$\mathcal{R}_{tok}^{add}(n_1,n_2) = \frac{\sum_{TP} D_{tok}(n_2,v)}{\sum_{F(n_2)} D_{tok}(n_2,v)} = \sum_{TP} P(v,n_2) \tag{4.13}$$

These simplified expressions correspond with the intuitions behind the token-based CRM given at the start of this section. Precision is the probability that a verb token picked at random from n_1's distribution also occurs with n_2; recall is the probability that a verb token picked at random from n_2's distribution also occurs with n_1.

4.1.1.3 Additive Mutual Information Based CRM

Mutual information (MI) allows us to capture the idea that a co-occurrence of low probability events is more informative than a co-occurrence of high probability events. For example, the co-occurrence of "have" and "dog" does not tell us very much about "dog" since "have" occurs very frequently with many different nouns. On the other hand, "dog" may have occurred less frequently as the object of "walk" but because "walk" occurs a lot less frequently in general, we learn more about "dog" from this co-occurrence.

More formally, following Church and Hanks (1989),Hindle (1990) and Lin (1998a), the degree of association between a noun and a verb is their pointwise mutual information:

$$D_{mi}(n,v) = I(n,v) = \log\left(\frac{P(v,n)}{P(v)P(n)}\right) \tag{4.14}$$

This expression tells us how much the probability that the verb is v is increased by knowing that the noun is n; or, conversely, how much the probability that the noun is n is increased by knowing that the verb is v. For our purposes, the base of the log taken is irrelevant since we will always be measuring a ratio.

Using MI as the association function means that a verb v will be considered to be a feature of noun n if the probability of their co-occurrence is greater than would be expected if verbs and nouns occurred independently. Further, more informative verb co-occurrences will add more to the sums in the calculation of precision and recall and hence have more weight.

The expressions for precision and recall in the additive MI-based CRM can be rewritten as follows:

$$\mathcal{P}_{mi}^{add}(n_1,n_2) = \frac{\sum_{TP} I(n_1,v)}{\sum_{F(n_1)} I(n_1,v)} \tag{4.15}$$

$$\mathcal{R}_{mi}^{add}(n_1,n_2) = \frac{\sum_{TP} I(n_2,v)}{\sum_{F(n_2)} I(n_2,v)} \tag{4.16}$$

4.1.2 Difference-Weighted Models

A possible criticism of additive models is that if two nouns have the same features they are considered identical regardless of whether the feature occurs with the same probability or to the same extent with each noun or not. In this section, we define a type of model that allows us to capture the difference in the extent to which each noun has each feature.

First, we could define a difference function that is the difference in extent, E, to which each noun is associated with each feature:

$$\text{difference}(n_1,n_2,v) = |E(n_1,v) - E(n_2,v)| \tag{4.17}$$

Since it is a difference, this value will be zero when the association scores are identical. This is the opposite to what we require in a weight for a similarity score and therefore we need to convert it into some kind of *shared extent* or *similarity* function. One possibility is to subtract the difference from the maximum possible score. For example, when the association function is the conditional probability of the verb given the noun, the maximum difference is 1 and therefore the shared extent function could be $(1 - \text{difference})$. This, however, has the undesirable property that

the score will lie on different scales for different association functions. An alternative approach is to make an analogy with the overall form of the distributional similarity measure we are defining. On this basis, we can attempt to define the similarity of two nouns in respect of an *individual* feature using the same principles that we use to define the similarity of two nouns in respect of all their features.

In order to do this we measure the precision and recall of an *individual* feature, v, of n_2 by n_1. First, we define an extent function, $E(n,v)$, which is the extent to which n_1 goes with v and which may be, but is not necessarily, the same as the association function $D(n,v)$. Possible extent functions will be discussed in the context of particular CRMs. Having defined this function, we can measure the precision and recall of individual features. The precision of an individual feature v retrieved by n_1 is the extent to which both nouns go with v divided by the extent to which n_1 goes with v. The recall of the retrieval of v by n_1 is the extent to which both nouns go with v divided by the extent to which n_2 goes with v.

$$\mathcal{P}(n_1,n_2,v) = \frac{min(E(n_1,v),E(n_2,v))}{E(n_1,v)} \tag{4.18}$$

$$\mathcal{R}(n_2,n_1,v) = \frac{min(E(n_1,v),E(n_2,v))}{E(n_2,v)} \tag{4.19}$$

Precision and recall of an individual feature, like precision and recall of a distribution, lie in the range [0,1]. We can now redefine precision and recall as follows:

$$\mathcal{P}^{dw}(n_1,n_2) = \frac{\sum_{TP} D(n_1,v) \times \mathcal{P}(n_1,n_2,v)}{\sum_{F(n_1)} D(n_1,v)} \tag{4.20}$$

$$\mathcal{R}^{dw}(n_1,n_2) = \frac{\sum_{TP} D(n_2,v) \times \mathcal{R}(n_1,n_2,v)}{\sum_{F(n_2)} D(n_2,v)} \tag{4.21}$$

Using precision and recall of individual features as weights in the definitions of precision and recall of a distribution captures the intuition that a retrieved verb is not simply right or wrong, it is right or wrong to a certain extent. In other words, features that are shared to a similar extent by each noun are considered more important in the calculation of distributional similarity.

We will now show how the precision and recall of an individual feature can be incorporated into type-based, token-based and MI-based CRMs. Given the association functions defined earlier, we can measure the precision and recall of retrieval of an individual feature in terms of tokens or in terms of MI. It does not make sense to measure this type of precision and recall in terms of types, since for an individual feature types and tokens are equivalent.

4.1.2.1 Difference-Weighted Type-Based CRM

In the difference-weighted type-based CRM, each verb co-occurrence type under consideration is weighted by the precision and recall of that individual feature. As already stated, there is no difference between type and token retrieval of an individual feature and hence the extent function used is the probabilistic association function:

$$E_{type}(n,v) = P(v|n) \tag{4.22}$$

Consequently, we can rewrite the expressions for the precision and recall of an individual feature as follows:

$$\mathcal{P}_{type}(n_1, n_2, v) = \frac{min(P(v|n_1), P(v|n_2))}{P(v|n_1)} \tag{4.23}$$

$$\mathcal{R}_{type}(n_1, n_2, v) = \frac{min(P(v|n_1), P(v|n_2))}{P(v|n_2)} \ (= \mathcal{P}_{type}(n_2, n_1, v)) \tag{4.24}$$

$\mathcal{P}_{type}(n_1, n_2, v)$ is 1 if $P(v|n_2) > P(v|n_1)$ and equal to $P(v|n_2)/P(v|n_1)$ otherwise. In the context of precision, it represents the extent to which the verb v retrieved by n_1 is correct. In the context of recall, it represents the extent to which the verb v required by n_2 is retrieved. Substituting into the expressions for precision and recall we get:

$$\mathcal{P}^{dw}_{type}(n_1, n_2) = \frac{\sum_{TP} D_{type}(n_1, v) \times \mathcal{P}_{type}(n_1, n_2, v)}{\sum_{F(n_1)} D_{type}(n_1, v)} = \frac{\sum_{TP} \frac{min(P(v|n_1), P(v|n_2))}{P(v|n_1)}}{|F(n_1)|} \tag{4.25}$$

$$\mathcal{R}^{dw}_{type}(n_1, n_2) = \frac{\sum_{TP} D_{type}(n_2, v) \times \mathcal{R}_{type}(n_1, n_2, v)}{\sum_{F(n_2)} D_{type}(n_2, v)} = \frac{\sum_{TP} \frac{min(P(v|n_2), P(v|n_1))}{P(v|n_2)}}{|F(n_2)|} \tag{4.26}$$

Note that this is different to the additive token-based model because, although every token is effectively considered in this model, tokens are not weighted equally. In this model, tokens are treated differently according to which type they belong to. The importance of the retrieval (or non-retrieval) of a single token depends on the proportion of the tokens for its particular type that it constitutes.

4.1.2.2 Difference-Weighted Token-Based CRM

In this model, each verb token under consideration is weighted by the precision or recall of tokens of its type. The extent function used is the probabilistic association function:

$$E_{token}(n, v) = P(v|n) \tag{4.27}$$

Hence, the precision and recall functions for individual features is the same as in the type-based CRM:

$$\mathcal{P}_{tok}(n_1, n_2, v) = \frac{min(P(v|n_1), P(v|n_2))}{P(v|n_1)} \tag{4.28}$$

$$\mathcal{R}_{tok}(n_1, n_2, v) = \mathcal{P}_{tok}(n_2, n_1, v) = \frac{min(P(v|n_1), P(v|n_2))}{P(v|n_2)} \tag{4.29}$$

Accordingly, we can derive expressions for precision and recall as follows:

$$\mathcal{P}^{dw}_{tok}(n_1, n_2) = \frac{\sum_{TP} D(n_1, v) \times \frac{min(E(n_1, v), E(n_2, v))}{E(n_1, v)}}{\sum_{F(n_1)} D(n_1, v)} \tag{4.30}$$

$$= \frac{\sum_{TP} P(v|n_1) \times \frac{min(P(v|n_1), P(v|n_2))}{P(v|n_1)}}{\sum_{F(n_1)} P(v|n_1)} \tag{4.31}$$

$$= \sum_{TP} min(P(v|n_1), P(v|n_2)) \tag{4.32}$$

$$\mathcal{R}_{tok}^{dw}(n_1,n_2) = \frac{\sum_{TP} D(n_2,v) \times \frac{min(E(n_2,v),E(n_1,v))}{E(n_2,v)}}{\sum_{F(n_2)} D(n_2,v)} \tag{4.33}$$

$$= \frac{\sum_{TP} P(v|n_2) \times \frac{min(P(v|n_2),P(v|n_1))}{P(v|n_2)}}{\sum_{F(n_2)} P(v|n_2)} \tag{4.34}$$

$$= \sum_{TP} min(P(v|n_2),P(v|n_1)) \tag{4.35}$$

$$= \mathcal{P}_{tok}^{dw}(n_1,n_2) \tag{4.36}$$

Note that although we have defined separate precision and recall functions, we have arrived at the same expression for both in this model. This is because difference-weighting takes into account the verb tokens not retrieved (for precision) and retrieved incorrectly (for recall). As a result, this model is symmetric, which may seem out of place in an asymmetric framework. However, we do not discount the possibility of a symmetric relationship between similar words, we simply do not assume it. The inclusion of this symmetric model, which has been derived from asymmetric principles, in our framework simply makes it a special case within the framework (we will see others later when we consider how to combine precision and recall). By allowing symmetry, we can thoroughly explore the relative merits of symmetry and asymmetry in similarity.

4.1.2.3 Difference-Weighted MI-Based CRM

We can consider the extent to which each noun occurs with each verb in terms of their MI, i.e.:

$$E_{mi}(n,v) = I(n,v) = \log\left(\frac{P(v,n)}{P(v)P(n)}\right) \tag{4.37}$$

This gives us the following expressions for the precision and recall of an individual feature:

$$\mathcal{P}_{mi}(n_1,n_2,v) = \frac{min(I(n_1,v),I(n_2,v))}{I(n_1,v)} \tag{4.38}$$

$$\mathcal{R}_{mi}(n_1,n_2,v) = \frac{min(I(n_1,v),I(n_2,v))}{I(n_2,v)} \tag{4.39}$$

Hence, the difference-weighted MI-based CRM gives us the following expressions for precision and recall of the retrieval of a co-occurrence distribution:

$$\mathcal{P}_{mi}^{dw}(n_1,n_2) = \frac{\sum_{TP} I(n_1,v) \times \frac{min(I(n_1,v),I(n_2,v))}{I(n_1,v)}}{\sum_{F(n_1)} I(n_1,v)} = \frac{\sum_{TP} min(I(n_1,v),I(n_2,v))}{\sum_{F(n_1)} I(n_1,v)} \tag{4.40}$$

$$\mathcal{R}_{mi}^{dw}(n_1,n_2) = \frac{\sum_{TP} I(n_2,v) \times \frac{min(I(n_2,v),I(n_1,v))}{I(n_2,v)}}{\sum_{F(n_2)} I(n_2,v)} = \frac{\sum_{TP} min(I(n_2,v),I(n_1,v))}{\sum_{F(n_2)} I(n_2,v)} \tag{4.41}$$

4.1.3 Combining Precision and Recall

We have, in the previous sections, been concerned with defining a pair of numbers that represent the similarity between two words. However, in applications, it will be necessary to compute a single number in order to determine neighbourhood or cluster membership. The classic way

to combine precision and recall in Information Retrieval is to compute the F-score, that is, the harmonic mean of precision and recall:

$$F = m_h(\mathcal{P}, \mathcal{R}) = \frac{2.\mathcal{P}.\mathcal{R}}{\mathcal{P}+\mathcal{R}} \tag{4.42}$$

The harmonic mean of two numbers A and B lies between the two numbers but is always substantially closer to the lower of the two numbers. For a given value of A+B, it is maximised when A=B. In other words, to achieve a reasonably high F score, both precision and recall must be reasonably high.

However, we do not wish to assume that a good substitute requires both high precision and high recall of the target distribution. It may be that, in some situations, the best noun to use in place of another noun is one that only retrieves correct co-occurrences (i.e. it is a high precision neighbour) or it may be one that retrieves all of the required co-occurrences (i.e. it is a high recall neighbour). The other factor in each case may play only a secondary role or no role at all.

We can retain generality and investigate whether high precision *or* high recall *or* high precision *and* high recall are required for high similarity by computing a weighted arithmetic mean of the harmonic mean and the weighted arithmetic mean of precision and recall:

$$m_h(\mathcal{P}(n_1,n_2), \mathcal{R}(n_1,n_2)) = \frac{2.\mathcal{P}(n_1,n_2).\mathcal{R}(n_1,n_2)}{\mathcal{P}(n_1,n_2)+\mathcal{R}(n_1,n_2)} \tag{4.43}$$

$$m_a(\mathcal{P}(n_1,n_2), \mathcal{R}(n_1,n_2)) = \beta.\mathcal{P}(n_1,n_2) + (1-\beta).\mathcal{R}(n_1,n_2) \tag{4.44}$$

$$sim(n_1,n_2) = \gamma.m_h(\mathcal{P}(n_1,n_2), \mathcal{R}(n_1,n_2)) + (1-\gamma).m_a(\mathcal{P}(n_1,n_2), \mathcal{R}(n_1,n_2)) \tag{4.45}$$

where both β and γ lie in the range [0,1]. The resulting similarity, $sim(n_1,n_2)$, will also lie in the range [0,1] where 0 is low and 1 is high. This formula can be used in combination with any of the models for precision and recall outlined earlier. Precision and recall can be computed once for every pair of words (and every model) whereas similarity depends on the values of β and γ. The generality allows us to investigate empirically the relative significance of the different terms and thus whether one (or more) might be omitted in future work. Table 4.1 summarizes some special parameter settings.

β	γ	Special Case
-	1	harmonic mean of precision and recall (F-score)
any	0	weighted arithmetic mean of precision and recall
1	0	precision
0	0	recall
0.5	0	unweighted arithmetic mean

Table 4.1: Table of special values of β and γ

4.1.4 Conclusion

In conclusion of this section, we have developed a framework based on the concept of co-occurrence retrieval (CR). Within this framework we have defined a number of models (CRMs) which allow

us to explore systematically three questions about similarity. First, does the difference in extent to which each noun takes each feature matter? Second, are some features inherently more salient than others due to their higher probability of occurrence or their higher information content? Third, is similarity between nouns necessarily a symmetric relationship?

The CRMs and the parameter settings therein provide different answers to these questions. Difference-weighted models contrast with additive models in considering the difference in extent to which each noun takes each feature. Type-based, token-based and MI-based CRMs contrast in the relative importance attached to features. Finally, a high precision neighbour is not necessarily a high recall neighbour (and, conversely, a high recall neighbour is not necessarily a high precision neighbour) and, therefore, we are not constrained to a symmetric relationship of similarity between words.

In the next section, we consider what effects the answers to these questions have on the nearest neighbours of a noun.

4.2 Neighbours Selected by Co-occurrence Retrieval Models

In this section, we first illustrate the neighbours obtained at four special parameter settings for each model within the framework. We then consider the relationship between parameter settings and any tendency for high or low frequency neighbours to be selected.

4.2.1 Neighbours of "hope"

As in Chapter 3, the neighbours of the word *hope* are selected from amongst the sample set of 2000 words (see Appendix A). The neighbours of *hope* are presented for the purposes of illustration only and we do not attempt to draw any conclusions from neighbour sets derived for just one word. Tables 4.2, 4.3, 4.4, 4.5 and 4.7 show the top ten neighbours of "hope" and their associated similarity scores for a number of different parameter settings in the additive type-based model, the additive token-based model, the additive MI-based model, the difference-weighted type-based model and the difference-weighted MI-based model respectively. Table 4.6 shows the top ten neighbours of *hope* and their associated similarity scores as selected by the distance-weighted token-based CRM. The same neighbours are selected in this model regardless of the parameter settings, since precision equals recall.

It should also be noted that all of the parameter settings for all of the CRMs (as well as the vast majority of other similarity measures), consider a word to be identical to itself and therefore a word will be its own closest neighbour[2]. Whether a word can be used as a neighbour of itself may be depend on the application and we will consider this issue more in later chapters. However, for the time being, we will consider it trivial to decide that the best word to use in place of a word is that word, and hence consider the word with the next highest similarity score to be the nearest neighbour.

4.2.2 Relationship between Parameter Settings and Frequency of Neighbours Selected

In this section, we examine the tendencies of the different models to select neighbours of particular frequencies using the technique described in Chapter 3 (Section 3.2.3). In particular we consider

[2]If there are other words which are identical to the word then it will be the joint closest neighbour.

($\beta = 1, \gamma = 0$) (precision)		($\beta = 0, \gamma = 0$) (recall)		($\beta = 0.5, \gamma = 0$) (arith. mean)		($\beta = n/a, \gamma = 1$) (harm. mean)	
stir	0.8	time	0.65	interest	0.42	dream	0.41
para	0.75	work	0.64	idea	0.42	chance	0.40
swig	0.75	people	0.64	feeling	0.42	confidence	0.39
talking	0.71	idea	0.63	stir	0.42	commitment	0.38
inkling	0.70	life	0.63	value	0.41	love	0.38
qualm	0.70	man	0.63	sense	0.41	concern	0.37
sparkle	0.69	thing	0.62	life	0.41	ambition	0.37
footing	0.68	interest	0.62	dream	0.41	sense	0.36
vocation	0.68	part	0.62	chance	0.40	feeling	0.36
grudge	0.67	power	0.61	power	0.40	expectation	0.36

Table 4.2: Neighbours of "hope" using additive type-based CRM

($\beta = 1, \gamma = 0$) (precision)		($\beta = 0, \gamma = 0$) (recall)		($\beta = 0.5, \gamma = 0$) (arith. mean)		($\beta = n/a, \gamma = 1$) (harm. mean)	
para	0.99	work	0.86	sense	0.82	sense	0.82
swig	0.98	thing	0.85	place	0.80	place	0.79
qualm	0.96	side	0.85	chance	0.79	chance	0.79
inkling	0.95	part	0.84	thing	0.78	thing	0.78
piss	0.95	life	0.84	feeling	0.76	feeling	0.75
hello	0.93	time	0.83	part	0.75	part	0.74
talking	0.93	position	0.83	idea	0.75	faith	0.74
errand	0.92	feeling	0.83	reason	0.75	idea	0.74
twinge	0.92	man	0.83	faith	0.74	reason	0.73
utmost	0.91	people	0.83	advantage	0.74	dream	0.73

Table 4.3: Neighbours of "hope" using additive token-based CRM

the relationship between the parameter settings in the model, the frequency of the target noun and the frequencies of the nearest neighbours. By the nature of precision and recall, we would expect to find that high recall neighbours occur in more distinct contexts than the target noun and high precision neighbours occur in less distinct contexts than the target noun. As distribution width and frequency are also correlated, we would expect this to result in precision selecting for lower frequency neighbours and recall selecting for higher frequency neighbours. When the harmonic mean is optimised, both precision and recall must be high which we would expect would lead to neighbours of a similar frequency to the target noun being selected. When the arithmetic mean is high, very high precision can compensate for very low recall and vice versa. Accordingly, we would expect these neighbour sets to be the least biased towards either high or low frequency neighbours.

In order to do this, we use the same technique described in Chapter 3 (Section 3.2.3). Recall that using this technique we compare neighbour sets (of size $k = 200$) which have been restricted to high or to low frequency nouns with neighbour sets which are unrestricted in frequency. We do

($\beta=1,\gamma=0$) (precision)		($\beta=0,\gamma=0$) (recall)		($\beta=0.5,\gamma=0$) (arith. mean)		($\beta=n/a,\gamma=1$) (harm. mean)	
bluff	0.52	dream	0.32	dream	0.29	dream	0.29
grudge	0.48	feeling	0.27	bluff	0.27	ambition	0.24
qualm	0.45	sense	0.25	grudge	0.26	confidence	0.23
vocation	0.44	interest	0.25	ambition	0.25	love	0.21
misgiving	0.41	claim	0.25	confidence	0.23	desire	0.20
twinge	0.38	spirit	0.24	vocation	0.23	sense	0.19
pretension	0.38	idea	0.23	misgiving	0.23	expectation	0.19
overtone	0.37	thought	0.23	qualm	0.22	doubt	0.19
self-confidence	0.37	man	0.23	confidence	0.22	suspicion	0.18
para	0.37	plan	0.23	pretension	0.21	fear	0.18

Table 4.4: Neighbours of "hope" using additive MI-based CRM

($\beta=1,\gamma=0$) (precision)		($\beta=0,\gamma=0$) (recall)		($\beta=0.5,\gamma=0$) (arith. mean)		($\beta=n/a,\gamma=1$) (harm. mean)	
regard	0.42	state	0.40	dream	0.30	dream	0.29
twinge	0.34	kind	0.39	commitment	0.29	commitment	0.29
heed	0.32	practice	0.38	concern	0.29	confidence	0.29
prospect	0.31	value	0.37	confidence	0.29	concern	0.28
chance	0.31	belief	0.37	feeling	0.29	love	0.28
dinner	0.29	world	0.37	love	0.28	ambition	0.26
confidence	0.29	feeling	0.36	memory	0.28	feeling	0.26
impression	0.28	power	0.36	belief	0.28	memory	0.26
doubt	0.28	society	0.36	value	0.28	expectation	0.26
laugh	0.28	industry	0.36	kind	0.28	chance	0.26

Table 4.5: Neighbours of "hope" using difference-weighted type-based CRM

this by computing the mean cosine or correlation between neighbour sets of restricted frequency and of unrestricted frequency. We then normalise this by dividing the average correlation with high frequency neighbour sets by the sum of the two correlation scores. This results in a score in the range [0,1] where 0 represents maximal tendency to select low frequency neighbours, 1 represents maximal tendency to select high frequency neighbours and 0.5 represents no tendency in either direction.

One question that remains is, how much deviation from 0.5 represents a significant tendency to select either high or low frequency nouns as neighbours? The standard deviations in the correlation scores observed all lay in the range [0,0.2]. By the Central Limit Theorem, a mean is always distributed normally and the variance in the distribution of the mean is the variance of the original distribution divided by the number of observations. Consequently, for the observed data, the maximum standard error (the square root of the variance of the mean) is 0.006. This means that a deviation from 0.5 of 0.014 is significant at the 99% level and a deviation from 0.5 of more than 0.0234 is statistically "impossible" if there is no bias towards high or low frequency nouns

All values of β and γ	
confidence	0.43
dream	0.42
feeling	0.42
desire	0.42
reason	0.42
vision	0.41
baby	0.41
opportunity	0.41
doubt	0.41
flavour	0.41

Table 4.6: Neighbours of "hope" using difference-weighted token-based CRM. The parameter settings are irrelevant in this model since precision is equal to recall.

($\beta = 1, \gamma = 0$) (precision)		($\beta = 0, \gamma = 0$) (recall)		($\beta = 0.5, \gamma = 0$) (arith. mean)		($\beta = n/a, \gamma = 1$) (harm. mean)	
grudge	0.23	dream	0.26	dream	0.23	dream	0.23
regard	0.21	spirit	0.18	ambition	0.18	ambition	0.18
dream	0.21	desire	0.18	confidence	0.18	confidence	0.18
ambition	0.19	feeling	0.17	doubt	0.16	doubt	0.15
confidence	0.19	opposition	0.17	desire	0.15	love	0.15
doubt	0.18	ambition	0.17	love	0.15	desire	0.15
self-confidence	0.17	interest	0.17	spirit	0.14	faith	0.14
misgiving	0.17	fear	0.17	feeling	0.14	suspicion	0.13
prospect	0.16	confidence	0.17	interest	0.14	expectation	0.13
bluff	0.16	memory	0.16	faith	0.14	sense	0.13

Table 4.7: Neighbours of "hope" using difference-weighted MI-based CRM

as neighbours. The deviations observed, however, are far greater than this, suggesting that chance has nothing to do with the selection of high or low frequency nouns as neighbours.

Table 4.8 summarises the tendencies of each model to select high or low frequency nouns. We will now, however, look at each model in more detail.

4.2.2.1 Additive Type-Based CRM

Figure 4.1 illustrates the tendencies of the additive type-based CRM to select high frequency neighbours at different parameter settings for high and low frequency target nouns.

In this model, there are very strong tendencies for high recall neighbours to be high frequency nouns and high precision neighbours to be low frequency nouns. The harmonic mean selects high frequency neighbours for high frequency nouns and low frequency neighbours for low frequency nouns. In other words, requiring both precision and recall to be fairly high (i.e. penalising for one being low) leads to neighbours of a similar frequency to the target noun.

CRM	Tendencies
additive type-based	high recall selects high frequency neighbours, high precision selects low frequency neighbours, high harmonic mean selects similar frequency neighbours
additive token-based	selects high frequency neighbours
additive MI-based	high recall selects high frequency neighbours, high precision selects low frequency neighbours, high harmonic mean selects similar frequency neighbours, high unweighted arithmetic mean appears unbiased
difference-weighted type-based	selects similar frequency neighbours
difference-weighted token-based	selects high frequency neighbours, bias is small for low frequency target nouns
difference-weighted MI-based	selects similar frequency neighbours

Table 4.8: Summary of tendencies to select high or low frequency neighbours by models in the framework

4.2.2.2 Additive Token-Based CRM

Figure 4.2 illustrates the tendencies of the additive token-based CRM to select high frequency neighbours at different parameter settings for high and low frequency target nouns.

In this model, there appears to be a tendency to select high frequency neighbours at all parameter settings, for both high and low frequency target nouns. This tendency is strongest at low values of β (high recall) but it is seen at fairly high values of β (only at $\beta = 1$ might we consider there to be no apparent bias in either direction). The tendency is also stronger for high frequency target nouns than for low frequency target nouns, which suggests that the bias is not towards *high* frequency target nouns as such, but towards nouns of a *higher* frequency than the target noun.

With regard to γ, there appears to be some tendency for it to increase preference for similar frequency neighbours but, again, it is not enough to counter the high frequency preferring nature of the model in general.

4.2.2.3 Additive MI-Based CRM

Figure 4.3 illustrates the tendencies of the additive MI-based CRM to select high frequency neighbours at different parameter settings for high and low frequency target nouns.

In this model, there is a fairly strong tendency for high recall to select for high frequency neighbours and for high precision to select for low frequency neighbours irrespective of the frequency of the target nouns. Further, the harmonic mean produces neighbour sets where the neighbours have a similar frequency to the target noun. These tendencies are similar to those seen in the ad-

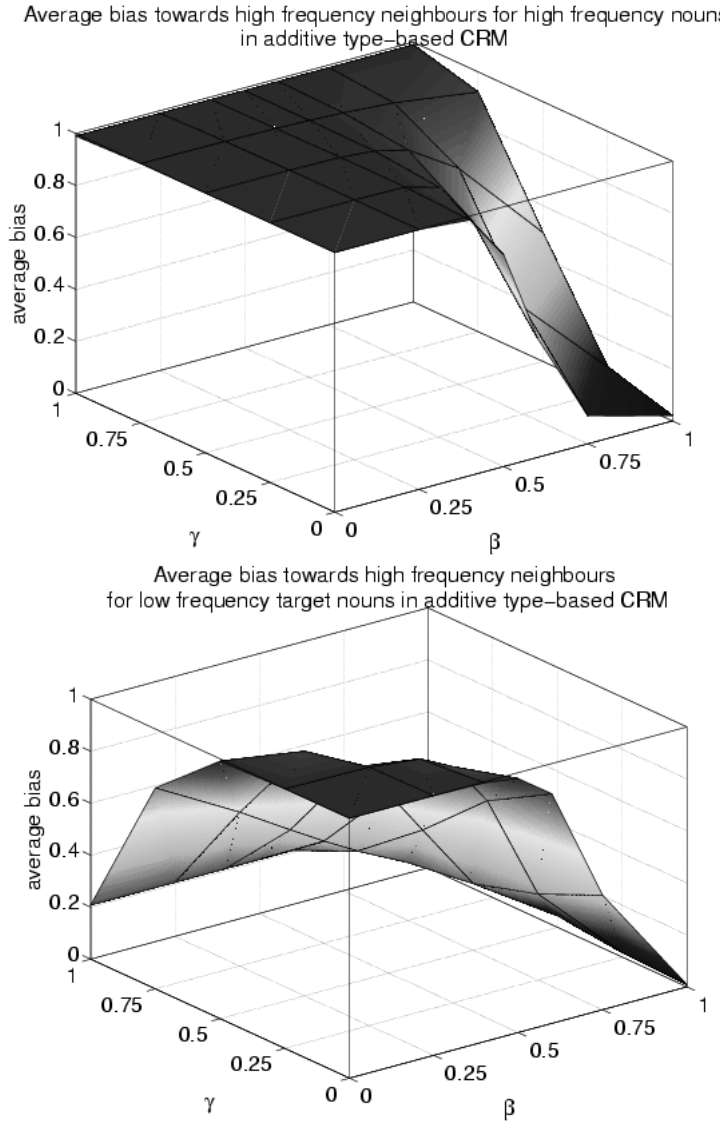

Figure 4.1: Tendencies of additive type-based CRM to select high frequency neighbours for high and low frequency target nouns

ditive type-based CRM but they are not as strong. However, in the additive MI-based CRM, an unweighted arithmetic mean produces neighbours sets with no apparent bias towards high or low frequency neighbours.

4.2.2.4 Difference-Weighted Type-Based CRM

Figure 4.4 illustrates the tendencies of the difference-weighted type-based CRM to select high frequency neighbours at different parameter settings for high and low frequency target nouns.

In this model, there is a strong tendency to select neighbours of a similar frequency to the target noun, on which the settings of the parameters β and γ have little impact.

4.2.2.5 Difference-Weighted Token-Based CRM

For this model, the average bias towards high frequency neighbours for high frequency target nouns was found to be 0.916 and the average bias towards high frequency neighbours for low frequency target nouns was 0.556. This suggests that there is some tendency towards selecting

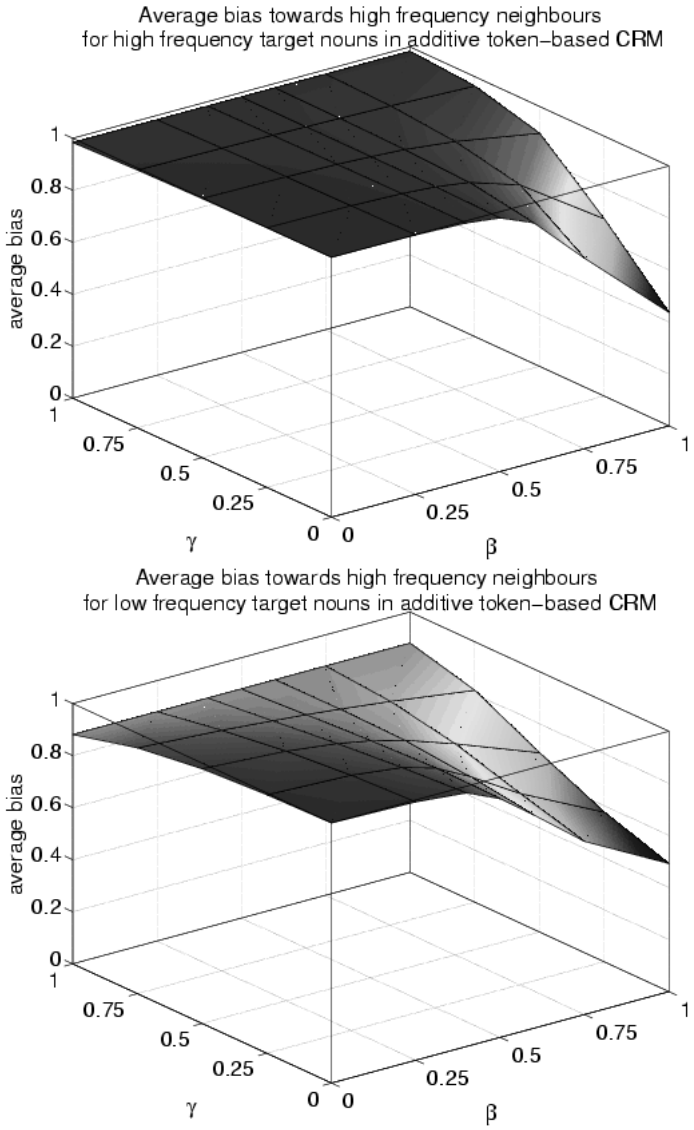

Figure 4.2: Tendencies for additive token-based CRM to select high frequency neighbours for high and low frequency target nouns

neighbours of higher frequency in this model, but is not as strong as in other models such as the additive token-based CRM.

4.2.2.6 Difference-Weighted MI-Based CRM

Figure 4.5 illustrates the tendencies of the difference-weighted MI-based model to select high frequency neighbours at different parameter settings for high and low frequency target nouns.

In this model, there is a strong tendency to select neighbours of a similar frequency to the target noun. However, the settings of the parameters β and γ do have some impact on this tendency. High precision reduces the tendency towards high frequency neighbours for high frequency target nouns; and high recall reduces the tendency towards low frequency neighbours for low frequency target nouns.

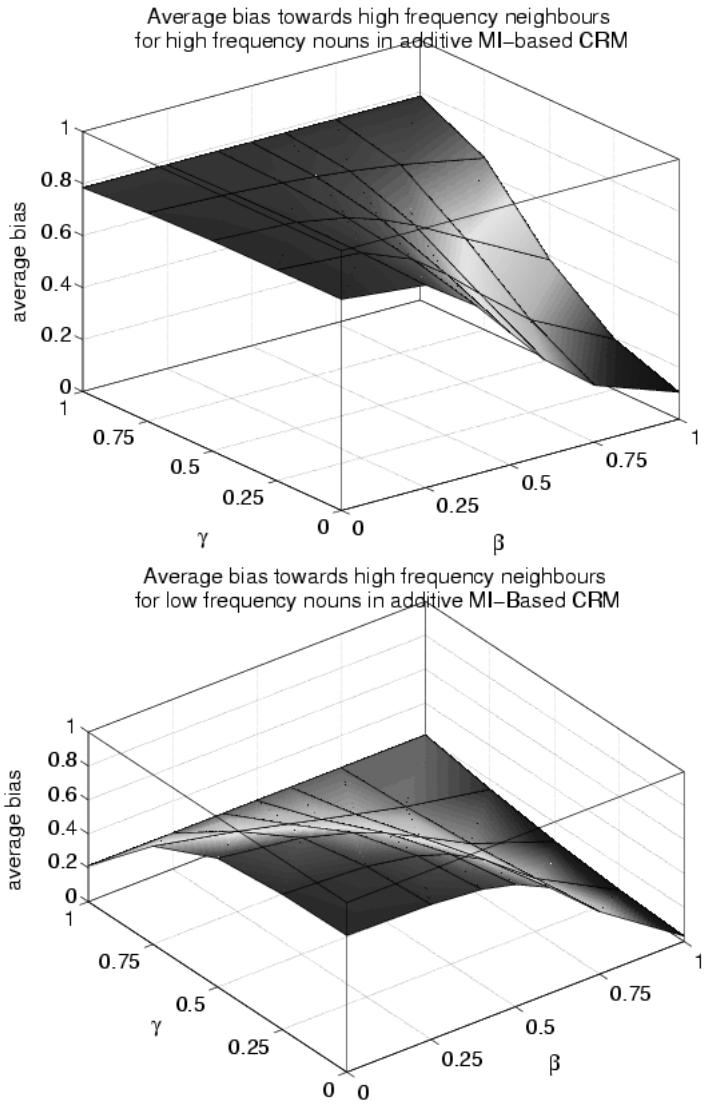

Figure 4.3: Tendencies for additive MI-based CRM to select high frequency neighbours for high and low frequency target nouns

4.2.3 Conclusion

In this section, we have looked at the neighbours produced by different models of the framework. In particular, we have examined the tendencies for different parameter settings to select for high or low frequency neighbours. We have seen that, in general, high precision produces a tendency towards low frequency neighbours, high recall produces a tendency towards high frequency neighbours and a high harmonic mean produces a tendency towards selecting neighbours of a similar frequency to the target noun. This is in line with our predictions and explanation at the start of this section.

However, the only time apparently unbiased neighbour sets are produced at the unweighted arithmetic mean is in the additive MI-based CRM. A possible reason for this is that the other models have inherent biases which strengthen or weaken the biases caused by the parameter settings. For example, the difference-weighted type-based and MI-based CRMs tend to select neighbours of a similar frequency to the target noun and the additive token-based CRM tends to select high

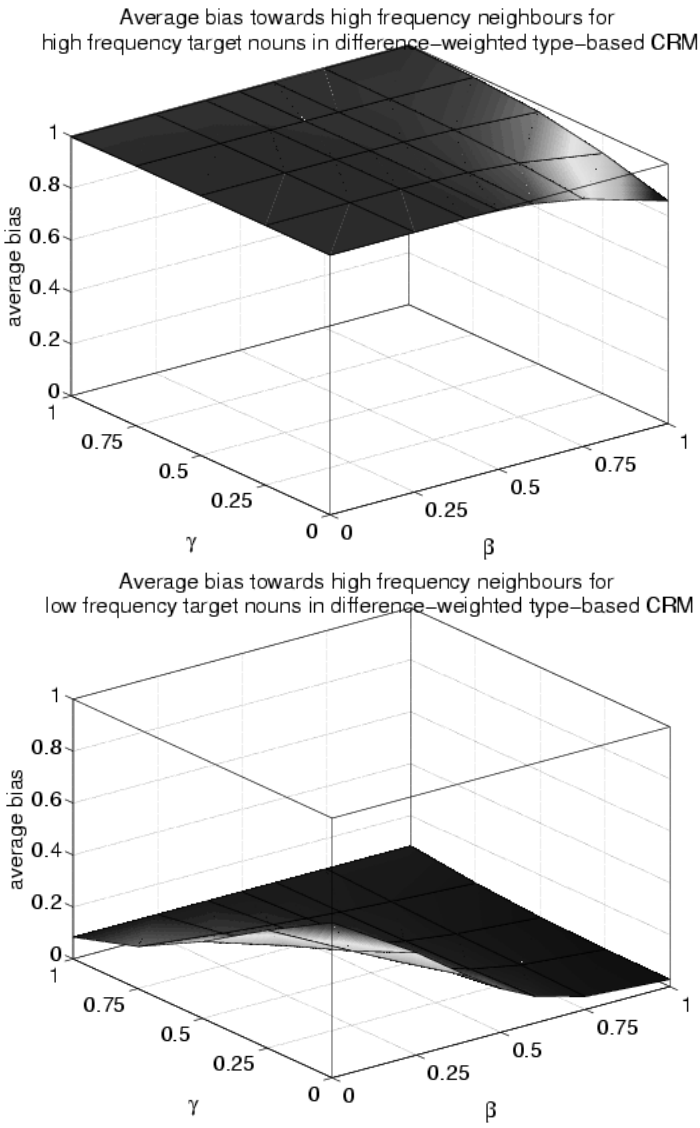

Figure 4.4: Tendencies for difference-weighted type-based CRM to select high frequency neighbours for high and low frequency target nouns

frequency neighbours for both high and low frequency target nouns.

4.3 Characterisations of Existing Measures

In this section, we present observations regarding existing similarity measures with reference to the CR framework offered in this chapter. Some of the existing measures can be directly derived or approximated in certain conditions using CR. For these, we offer mathematical proofs of the relationships. For other measures, which cannot be derived from the CR framework, we consider which model and parameter settings most closely simulate them. We do this using the neighbour set comparison technique[3] throughout this work. In these cases we calculated mean similarities between neighbour sets over the 1000 high frequency nouns and over the 1000 low frequency

[3]Recall that we take the 200 nearest neighbours according to each measure (or model and parameter setting) and compute the average correlation between rank scores.

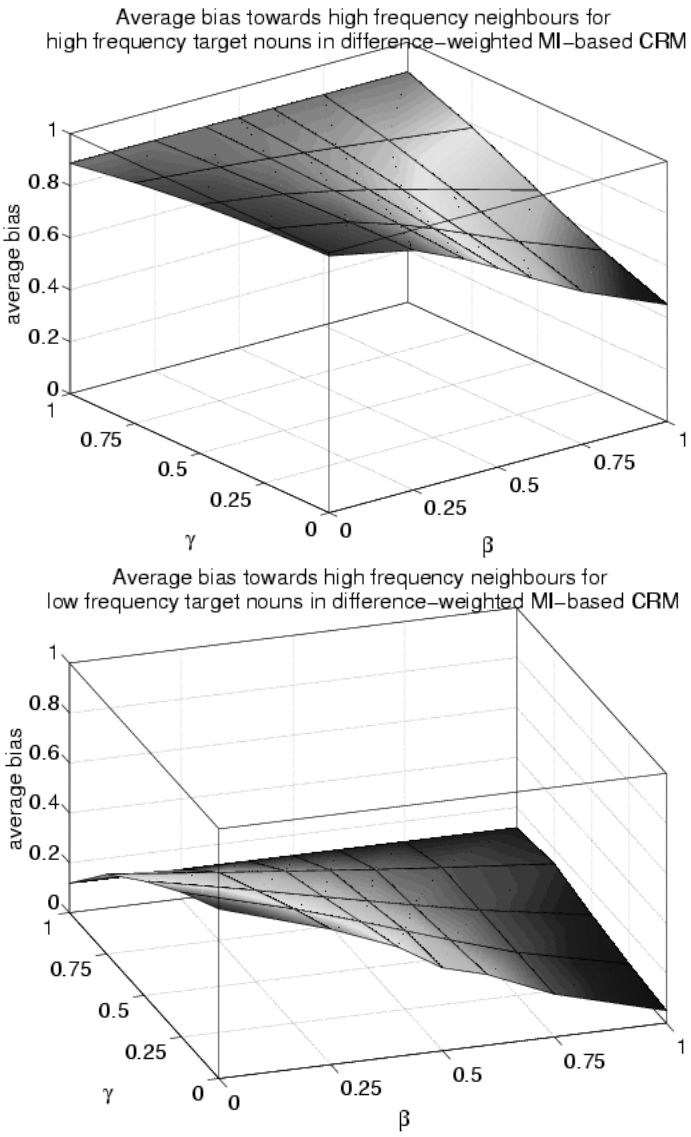

Figure 4.5: Tendencies for difference-weighted MI-based model to select high frequency neighbours for high and low frequency target nouns

nouns. Again, we need to consider what differences between means can be considered statistically significant. The maximum standard deviation in the correlation scores observed is 0.2, which corresponds to a standard error in a normally distributed mean of 0.006. The difference between two such means will also be normally distributed with standard deviation around 0.009. This corresponds to a difference of 0.01 being significant at the 90% level, a difference of 0.02 being significant at the 99% level and a difference of 0.035 being statistically "impossible" by chance. Accordingly, we will not refer to levels of significance for differences that are greater than this.

We will now consider each measure from Chapter 3 in turn. Table 4.9, however, summarizes the observations about each measure.

4.3.1 The L_1 Norm

Recall that the L_1 Norm is defined as:

Measure	Observations
$dist_{L_1}$	distance equivalent of the difference-weighted token-based CRM
$dist_{L_2}$	most closely simulated by the difference-weighted token-based CRM
sim_{cos}	most closely simulated by the difference-weighted token-based CRM
sim_{τ}	approximated by recall in the additive type-based CRM
sim_{dice}	equivalent to $\gamma = 1$ in the additive type-based CRM
sim_{jacc}	produces identical neighbour sets to $\gamma = 1$ in the additive type-based CRM
$dist_{js}$	most closely simulated by the difference-weighted token-based CRM
$dist_{\alpha 1}$	most closely simulated using high recall parameter settings in the additive token-based CRM
$dist_{\alpha 2}$	most closely simulated using high precision parameter settings in the additive token-based CRM
sim_{cp}	most closely simulated by high recall parameter settings in the additive token-based CRM
$sim_{cp_{rev}}$	most closely simulated by a range of parameter settings in the difference-weighted MI-based CRM
sim_{hind+}	produces identical neighbour sets to recall in the difference-weighted MI-based CRM
sim_{lin}	very highly similar to $\gamma = 1$ in the additive MI-based CRM
$sim_{jacc_{mi}}$	closely simulated by the $\gamma = 1$ setting in a number of models including the additive MI-based CRM

Table 4.9: Summary of observations about existing measures with reference to the CR Framework

$$dist_{L_1}(n_1, n_2) = \sum_v |P(v|n_1) - P(v|n_2)| \tag{4.46}$$

Noting the algebraic equivalence $A + B - |A - B| = 2 \times min(A, B)$ and using basic probability theory we can rewrite the L_1 Norm as follows:

$$\begin{aligned} L_1(n_1, n_2) &= \sum_v |P(v|n_1) - P(v|n_2)| && (4.47)\\ &= \sum_v (P(v|n_1) + P(v|n_2) - 2 \times min(P(v|n_1), P(v|n_2)) && (4.48)\\ &= \sum_v P(v|n_1) + \sum_v P(v|n_2) - 2 \times \sum_v min(P(v|n_1), P(v|n_2)) && (4.49)\\ &= 2 - 2 \times \sum_v min(P(v|n_1), P(v|n_2)) && (4.50) \end{aligned}$$

However, $min(P(v|n_1), P(v|n_2)) > 0$ if and only if $v \in TP$. Hence:

$$L_1(n_1, n_2) = 2 - 2 \times \sum_{TP} min(P(v|n_1), P(v|n_2)) \tag{4.51}$$

$$= \quad 2 - 2 \times sim^{dw}_{tok}(n_1, n_2) \tag{4.52}$$

In other words, the L_1 Norm is directly related to the difference-weighted token-based CRM. The constant and multiplying factor are required since the CRM defines a similarity in the range [0,1] whereas the L_1 Norm defines a distance in the range [0,2] (where 0 distance is equivalent to 1 on the similarity scale).

4.3.2 The L_2 Norm

We do not expect to find that $dist_{L_2}$ is exactly simulated by any CRM since it is quadratic in the differences between feature values. However, as discussed above, we can analyse the relationship between the L_2 Norm and the CR framework empirically by comparing the neighbour sets produced by each. Table 4.10 shows the optimal parameter settings and optimal similarities with this measure for each of the CRMs, as calculated using the neighbour set comparison technique first described in Chapter 3.

CRM	Target noun frequency					
	high			low		
	γ	β	Sim.	γ	β	Sim.
add-type	0.25	0.4	0.36 (0.12)	0.0	0.0	0.35 (0.10)
add-tok	0.25	0.0	0.36 (0.10)	0.25	0.0	0.35 (0.08)
add-mi	0.0	0.0	0.43 (0.14)	0.0	0.0	0.42 (0.12)
diff-type	0.0	0.0	0.48 (0.19)	0.0	0.0	0.35 (0.12)
diff-tok	-	-	**0.64 (0.10)**	-	-	**0.46 (0.15)**
diff-mi	0.0	0.0	0.43 (0.15)	0.0	0.0	0.36 (0.12)

Table 4.10: Mean optimal similarities between CRMs and $dist_{L_2}$. Figures in brackets are standard deviations.

As can be seen from Table 4.10, $dist_{L_2}$ is not closely simulated by any of the CRMs. The closest approximation, for both high and low frequency target nouns, is from the symmetric difference-weighted token-based CRM. A relationship between this measure and this model might have been expected since this model exactly simulates the L_1 Norm and the L_1 Norm comes from the same family of measures as the L_2 Norm. However, as also seen in Chapter 3, the similarity between neighbour sets derived using the L_1 Norm and using the L_2 Norm is not as great as is observed between other, less obviously related, pairs of measures (such as the L_1 Norm and the Jensen-Shannon Divergence Measure).

4.3.3 Cosine Measure

Again, there is no obvious similarity between the form of the cosine measure and any of the CRMs. Table 4.11, however, shows the optimal parameter settings and optimal similarities with this measure for each of the CRMs, as calculated using the neighbour set comparison technique.

As can be seen from Table 4.11, sim_{cos} is not closely simulated by any of the CRMs. The closest approximation, for both high and low frequency target nouns, is from the symmetric difference-weighted token-based CRM.

CRM	Target noun frequency					
	high			low		
	γ	β	Sim.	γ	β	Sim.
add-type	0.5	0.6	0.32 (0.13)	0.5	0.5	0.29 (0.12)
add-tok	0.75	0.8	0.36 (0.10)	0.75	0.6	0.43 (0.14)
add-mi	0.5	0.3	0.35 (0.15)	0.5	0.3	0.36 (0.15)
diff-type	0.25	0.5	0.31 (0.13)	0.0	0.0	0.36 (0.16)
diff-tok	-	-	**0.65 (0.14)**	-	-	**0.69 (0.13)**
diff-mi	0.25	0.4	0.35 (0.15)	0.0	0.0	0.36 (0.16)

Table 4.11: Mean optimal similarities between CRMs and sim_{cos}. Figures in brackets are standard deviations.

4.3.4 Kendall's τ Coefficient

Recall the description of Kendall's τ coefficient from Chapter 3. For every pair of co-occurrence types (v_i, v_j), we consider the signs of the quantities $P(v_i|n_1) - P(v_j|n_1)$ and $P(v_i|n_2) - P(v_j|n_2)$. If both signs are the same, then the verb pair is a concordance; if the signs are different, the pair is a discordance; and if either quantity is zero, then the pair is a tie. $sim_\tau(n_1, n_2)$ can then be estimated using:

$$\frac{\text{number of observed concordances} - \text{number of observed discordances}}{|V|(|V|-1)} \tag{4.53}$$

where V is the set of all possible co-occurrence types.

We can, however, consider this measure in terms of type-based co-occurrence retrieval. Each co-occurrence type may be a feature of both nouns, of one noun but not the other or of neither noun. Due to the nature of co-occurrence data we would expect a large number of co-occurrence types to be observed with neither noun. Accordingly, pairs of verbs where at least one verb is not observed with either noun will dominate the expression for the τ coefficient. In the case where neither verb goes with the target noun, there will be a tie and this will be the same for all potential neighbours. However, if one verb (but not the other) goes with the target noun, whether we have a concordance or discordance will most often depend on whether that verb co-occurs with the potential neighbour. In the small number of cases where both verbs co-occur with the target noun, the result of concordance or discordance will depend on the extent to which each verb is seen with each noun. However, this will be for a very small minority of the verb pairs and consequently the value of the τ coefficient will largely depend on the number of shared co-occurrence types, i.e., the size of the set of true positives, $|TP|$.

Now let us consider again the recall in the additive type-based CRM:

$$\mathcal{R}^{add}_{type}(n_1, n_2) = \frac{\sum_{TP} D_{type}(n_2, v)}{\sum_{F(n_2)} D_{type}(n_2, v)} = \frac{|TP|}{|F(n_2)|} \tag{4.54}$$

For a given target noun, n_2, the value of $|F(n_2)|$ will be a constant. This means that the difference in recall values between potential neighbours, for a given target noun, depends only on $|TP|$. As a consequence, we would expect the neighbour sets of the τ coefficient to be closely simulated using a parameter setting of $(\gamma = 0, \beta = 0)$ (i.e. recall) in the additive type-based CRM. How close

the approximation is will depend largely on how many verbs co-occur with neither noun. Table 4.12 summarises the optimal parameter settings and optimal mean similarities between neighbour sets of sim_τ and each of the CRMs.

CRM	Target noun frequency					
	high			low		
	γ	β	Sim.	γ	β	Sim.
add-type	0.0	0.1	**0.9992 (0.0006)**	0.0	0.25	**0.9984 (0.0043)**
add-tok	0.0	0.0	0.89 (0.07)	0.0	0.0	0.92 (0.07)
add-mi	0.0	0.0	0.72 (0.06)	0.0	0.0	0.66 (0.09)
diff-type	0.0	0.0	0.74 (0.11)	0.0	0.0	0.31 (0.09)
diff-tok	-	-	0.43 (0.13)	-	-	0.23 (0.09)
diff-mi	0.0	0.0	0.52 (0.09)	0.0	0.0	0.38 (0.10)

Table 4.12: Mean optimal similarities between CRMs and sim_τ. Figures in brackets are standard deviations.

As predicted, there is very high similarity between neighbour sets of the τ coefficient and high recall settings of the additive type-based CRM. Figure 4.6 shows how similarity between the neighbour sets varies with the parameters β and γ for this CRM.

As can be seen from Figure 4.6, optimal similarity between the neighbour sets is very close to 1 for both high and low frequency nouns. The optimal mean similarity values are 0.9992 at $(\gamma = 0, \beta = 0.1)$ for high frequency nouns and 0.9984 at $(\gamma = 0, \beta = 0.25)$ for low frequency nouns. The small amount of precision incorporated at these settings may approximate the adjustment required when both verbs occur with one of the nouns. However the values at $(\gamma = 0, \beta = 0)$, which are 0.9962 and 0.9875 respectively for high and low frequency target nouns, are very close to the optimal values (the difference for low frequency target nouns is statistically significant at the 90% level). This suggests that the τ coefficient is almost completely determined by the number of shared co-occurrence types, $|TP|$.

4.3.5 Dice and Jaccard's Coefficient

Recall that the definition of the Dice Coefficient is:

$$sim_{dice}(n_1, n_2) = \frac{2.|F(n_1) \cap F(n_2)|}{|F(n_1)| + |F(n_2)|} \text{ where } F(n) = \{v : P(v|n) > 0\} \tag{4.55}$$

However, as shown below, this formula is equivalent to a special case in our framework: the harmonic mean of precision and recall (or F-score) using the additive type-based CRM.

$$m_h(\mathcal{P}^{add}_{type}(n_1, n_2), \mathcal{R}^{add}_{type}(n_1, n_2)) = \frac{2.\mathcal{P}^{add}_{type}(n_1, n_2).\mathcal{R}^{add}_{type}(n_1, n_2)}{\mathcal{P}^{add}_{type}(n_1, n_2) + \mathcal{R}^{add}_{type}(n_1, n_2)} \tag{4.56}$$

$$= \frac{2.\frac{|TP|}{|F(n_1)|}.\frac{|TP|}{|F(n_2)|}}{\frac{|TP|}{|F(n_1)|} + \frac{|TP|}{|F(n_2)|}} \tag{4.57}$$

$$= \frac{2.|TP|.|TP|}{|TP|.|F(n_1)| + |TP|.|F(n_2)|} \tag{4.58}$$

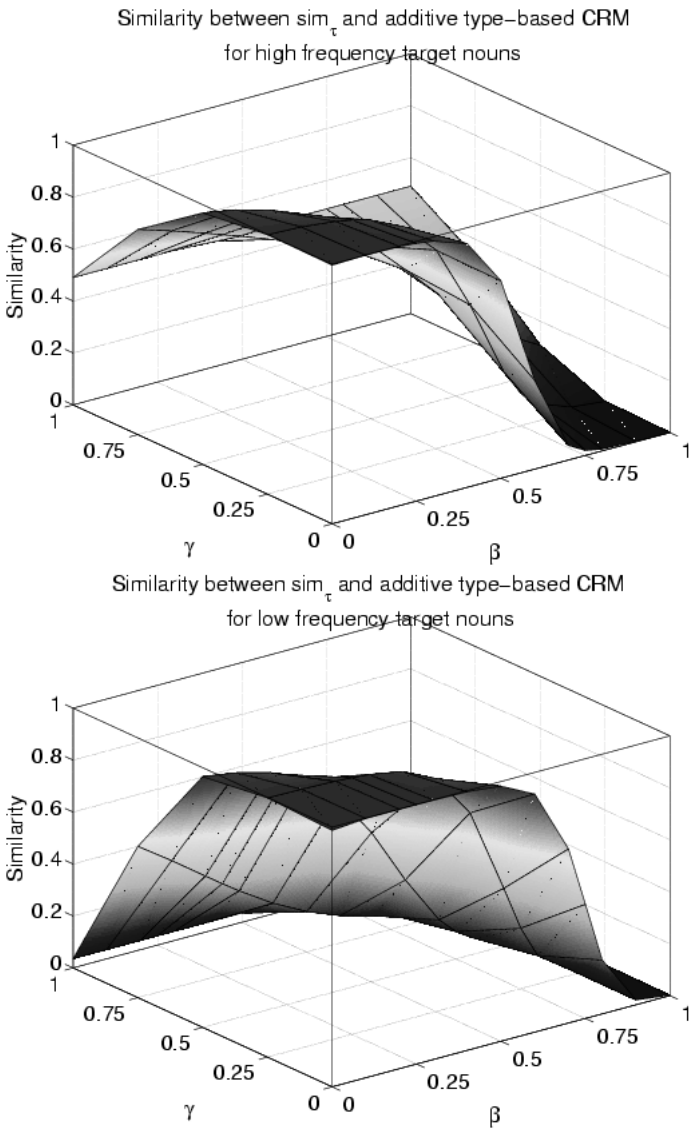

Figure 4.6: Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive type-based CRM and of sim_τ

$$= \frac{2.|TP|}{|F(n_1)|+|F(n_2)|} \tag{4.59}$$

$$= \frac{2.|F(n_1)\cap F(n_2)|}{|F(n_1)|+|F(n_2)|} \tag{4.60}$$

$$= \text{sim}_{\text{dice}}(n_1,n_2) \tag{4.61}$$

Accordingly, when γ is set to 1 in the additive type-based CRM, the Dice Coefficient is exactly replicated. Since we have already seen that Jaccard's Coefficient and the Dice Coefficient produce identical neighbour sets, it goes without saying that the neighbour sets produced by the additive type-based CRM at these settings will also be identical to those produced by Jaccard's Coefficient.

The relationship between the framework and these existing measures of distributional similarity can also be viewed empirically by comparing the neighbours produced by Jaccard's Coefficient/ the Dice Coefficient and those produced by the additive type-based CRM at different parameter settings. The mean similarities, as calculated using the neighbour set comparison technique, are

illustrated in Figure 4.7. As predicted, the similarity is exactly 1 at $\gamma = 1$ for both high and low frequency target nouns. As γ decreases so does the similarity between the two measures. At lower values of γ, high recall leads to higher similarity between the neighbour sets for high frequency nouns and high precision leads to higher similarity between the neighbour sets for low frequency nouns.

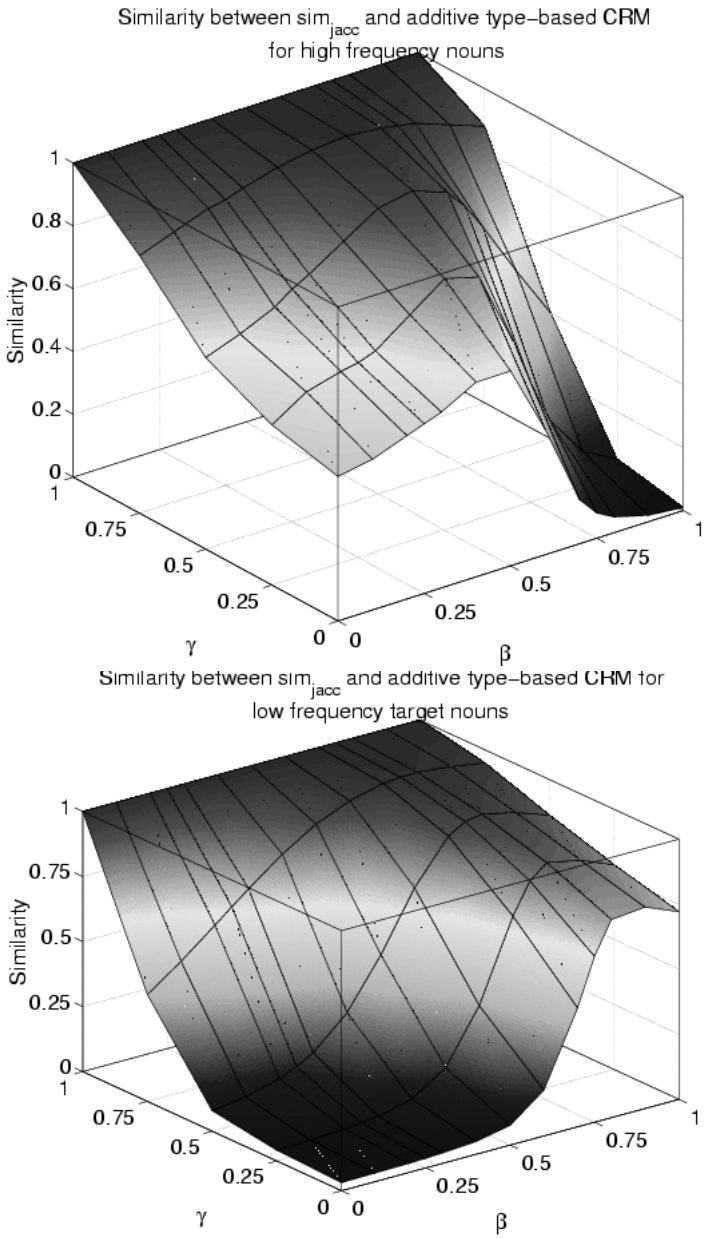

Figure 4.7: Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive type-based CRM and of sim_{jacc}.

4.3.6 Jensen-Shannon Divergence Measure

Having already noted the similarity between the Jensen-Shannon Divergence Measure and the L_1 Norm in Chapter 3 and between the L_1 Norm and the difference-weighted token-based CRM in this chapter, we expect to find that this measure is most similar to the difference-weighted token-based

CRM.

Table 4.13 shows the optimal mean similarities between neighbour sets derived using each of the CRMs and $dist_{js}$. As usual, the mean similarities are calculated separately for the 1000 high frequency target nouns and the 1000 low frequency target nouns.

CRM	Target noun frequency					
	high			low		
	γ	β	Sim.	γ	β	Sim.
add-type	0.5	0.5	0.65 (0.14)	0.75	0.1	0.52 (0.15)
add-tok	0.75	0.2	0.67 (0.09)	0.75	0.2	0.0.59 (0.11)
add-mi	0.5	0.1	0.67 (0.15)	0.5	0.3	0.57 (0.16)
diff-type	0.25	0.4	0.64 (0.16)	0.0	0.0	0.60 (0.17)
diff-tok	-	-	**0.93 (0.03)**	-	-	**0.85 (0.06)**
diff-mi	0.25	0.1	0.64 (0.16)	0.0	0.1	0.55 (0.17)

Table 4.13: Mean optimal similarities between CRMs and $dist_{js}$. Figures in brackets are standard deviations.

As can be seen from Table 4.13, neighbour sets derived using $dist_{js}$, for both high and low frequency target nouns, are significantly more similar to those derived using the difference-weighted token-based CRM than using any other CRM.

4.3.7 The α-skew Divergence Measure

Due to the form of the α-skew divergence measure, we do not expect any of the CRMs to exactly simulate it. However, we note that this measure does take into account the differences between the probabilities of co-occurrences in each distribution (as a log ratio) and therefore we might expect that it will be fairly closely simulated by the difference-weighted token-based CRM. We also might expect there to be some similarity with the MI-based models since $dist_{\alpha}$ is an approximation to the Information-Theoretic concept of relative entropy. Lastly, we recall that the α-skew divergence measure is asymmetric. $dist_{\alpha 1}(n_1, n_2)$ measures the cost of using the distribution of n_1 instead of n_2 and is calculated over the verbs which occur with n_2. $dist_{\alpha 2}(n_1, n_2)$ measures the cost of using the distribution of n_2 instead of n_1 and is calculated over the verbs which co-occur with n_1. As such, we might expect that $dist_{\alpha 1}$ will be a high recall measure, since recall is calculated over the co-occurrences of n_2, and $dist_{\alpha 2}$ will be a high precision measure, since precision is calculated over the co-occurrences of n_1.

Table 4.14 shows the optimal mean similarities between neighbour sets derived using each of the CRMs and $dist_{\alpha 1}$. As usual, the mean similarities are calculated separately for the 1000 high frequency target nouns and the 1000 low frequency target nouns.

From the results in Table 4.14 we observe the following. None of the CRMs are able to simulate $dist_{\alpha 1}$ very closely. Further, the differences between the models are fairly small. For high frequency nouns, the best performing models are the difference-weighted token based model and the additive token-based model. For low frequency nouns, the best performing models are the additive MI-based model and the additive token-based model. Most notably, all of the asymmetric models get closest at high levels of recall for both high and low frequency nouns. Figure 4.8

CRM	Target noun frequency					
	high			low		
	γ	β	Sim.	γ	β	Sim.
add-type	0.25	0.4	0.74 (0.14)	0.5	0.3	0.66 (0.15)
add-tok	0.5	0.0	0.77 (0.10)	0.5	0.0	0.67 (0.10)
add-mi	0.0	0.0	0.76 (0.14)	0.25	0.1	**0.73 (0.14)**
diff-type	0.0	0.25	0.71 (0.15)	0.0	0.0	0.53 (0.13)
diff-tok	-	-	**0.80 (0.09)**	-	-	0.58 (0.11)
diff-mi	0.0	0.0	0.67 (0.14)	0.0	0.0	0.60 (0.13)

Table 4.14: Mean optimal similarities between CRMs and $dist_{\alpha 1}$. Figures in brackets are standard deviations.

illustrates the variation in mean similarity between neighbour sets with the parameters β and γ for the additive token-based model. For high frequency target nouns, optimal similarity of 0.77 is at $(\gamma = 0, \beta = 0.25)$, which is significantly higher than 0.69 at $(\gamma = 1, \beta)$ and 0.10 at $(\gamma = 0, \beta = 1)$. For low frequency target nouns, optimal similarity of 0.67 is at $(\gamma = 0.5, \beta = 0)$, which is considerably higher than 0.44 at $(\gamma = 1, \beta)$ and 0.15 at $(\gamma = 0, \beta = 1)$. From this it appears that $dist_{\alpha 1}$ has high recall CR characteristics. We will now contrast this with $dist_{\alpha 2}$, which the reader will recall is equivalent to $dist_{\alpha 1}$ with its arguments reversed.

Table 4.15 shows the optimal mean similarities between neighbour sets derived using each of the CRMs and $dist_{\alpha 2}$. As usual, the mean similarities are calculated separately for the 1000 high frequency target nouns and the 1000 low frequency target nouns.

CRM	Target noun frequency					
	high			low		
	γ	β	Sim.	γ	β	Sim.
add-type	0.5	0.7	0.56 (0.13)	0.25	0.75	0.40 (0.11)
add-tok	0.5	0.9	0.67 (0.07)	0.25	0.9	**0.77 (0.10)**
add-mi	0.5	0.4	0.56 (0.14)	0.0	0.8	0.43 (0.11)
diff-type	0.25	0.7	0.51 (0.12)	0.0	0.25	0.39 (0.13)
diff-tok	-	-	**0.74 (0.10)**	-	-	0.59 (0.15)
diff-mi	0.5	0.6	0.55 (0.14)	0.0	0.6	0.38 (0.13)

Table 4.15: Mean optimal similarities between CRMs and $dist_{\alpha 2}$. Figures in brackets are standard deviations.

The optimal performance figures for simulating $dist_{\alpha 2}$ are similar to those for simulating $dist_{\alpha 1}$. However, there are larger differences now between the results for high and low frequency target nouns. The most consistently high performing model in this case is the additive token-based CRM. Further, as predicted, there is a reversal in the values of β at which the optimal performance is obtained. For the majority of CRMs, neighbours of both high and low frequency target nouns are required to have high precision to simulate $dist_{\alpha 2}$. This is illustrated in Figure 4.9 for the additive token-based CRM. However, we observe that for high frequency target nouns in particular, the effect is not an exact reflection of what was seen for $dist_{\alpha 1}$. Here, it would seem that some

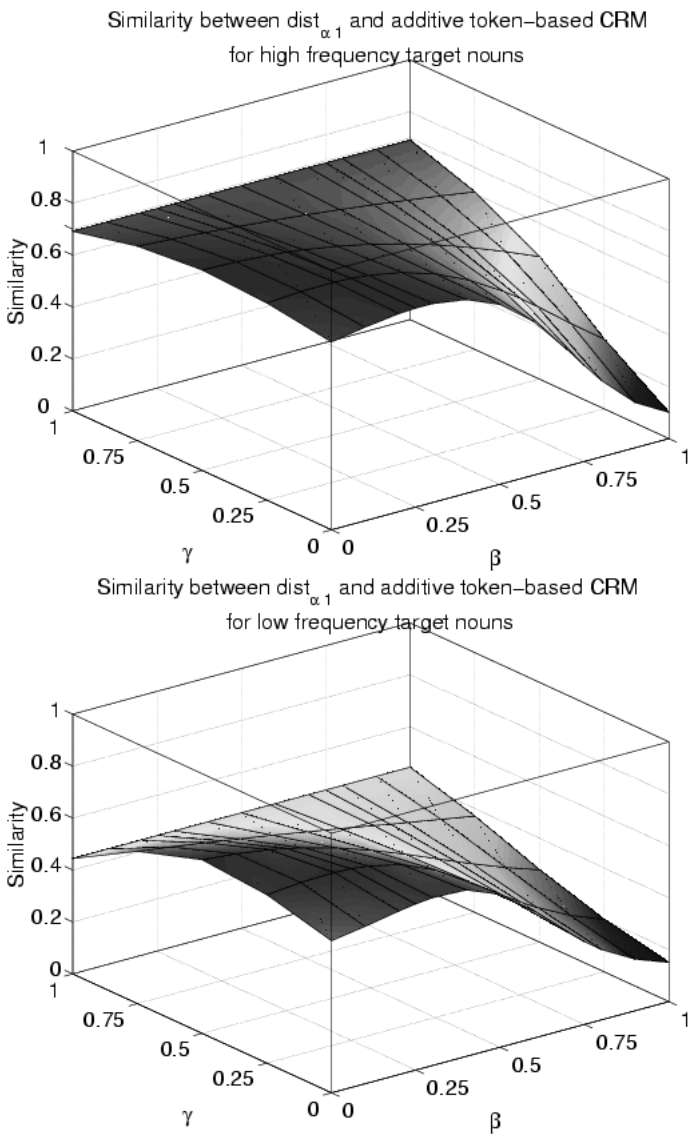

Figure 4.8: Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive token-based CRM and of $dist_{\alpha 1}$

amount of recall is required before the neighbour sets are similar. Sacrificing all recall for very high precision leads to a considerable drop in performance. This is likely to be because a high precision neighbour can be a noun that has only occurred in one distinct context. This noun will have a precision score of 1 for the retrieval of any other noun that occurs in that context. There is no parallel for this in terms of recall, since we do not expect to find a noun that co-occurs with every verb.

4.3.8 Confusion Probability

Table 4.16 shows the optimal mean similarities between neighbour sets derived using each of the CRMs and sim_{cp}. As usual, the mean similarities are calculated separately for the 1000 high frequency target nouns and the 1000 low frequency target nouns.

Although not very closely simulated by any of the CRMs, the confusion probability is most closely simulated by the additive token-based CRM. Figure 4.10 shows how the mean similarity

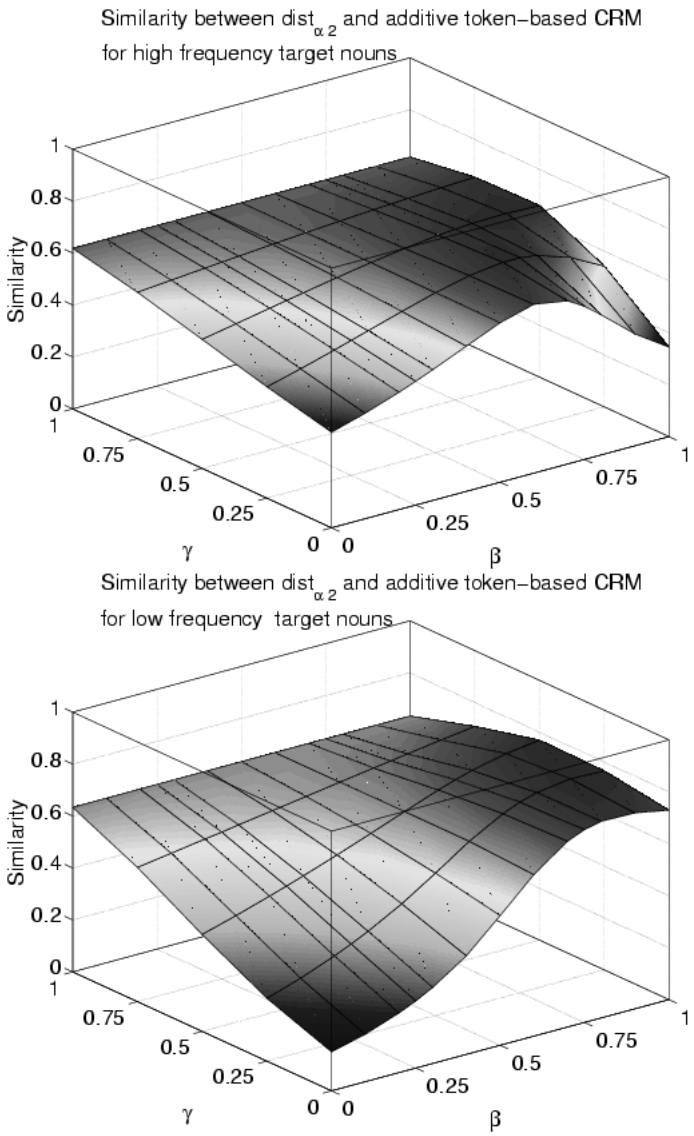

Figure 4.9: Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive token-based CRM and of $dist_{\alpha 2}$

CRM	Target noun frequency					
	high			low		
	γ	β	Sim.	γ	β	Sim.
add-type	0.0	0.2	0.76 (0.07)	0.0	0.0	0.70 (0.07)
add-tok	0.0	0.25	**0.78 (0.06)**	0.0	0.25	**0.74 (0.06)**
add-mi	0.0	0.0	0.62 (0.08)	0.0	0.0	0.52 (0.10)
diff-type	0.0	0.0	0.50 (0.10)	0.0	0.0	0.28 (0.08)
diff-tok	-	-	0.49 (0.08)	-	-	0.30 (0.09)
diff-mi	0.0	0.0	0.48 (0.09)	0.0	0.0	0.38 (0.09)

Table 4.16: Mean optimal similarities between CRMs and sim_{cp}. Figures in brackets are standard deviations.

between neighbour sets varies with β and γ for this measure and this CRM. It would appear that the confusion probability is a high recall measure.

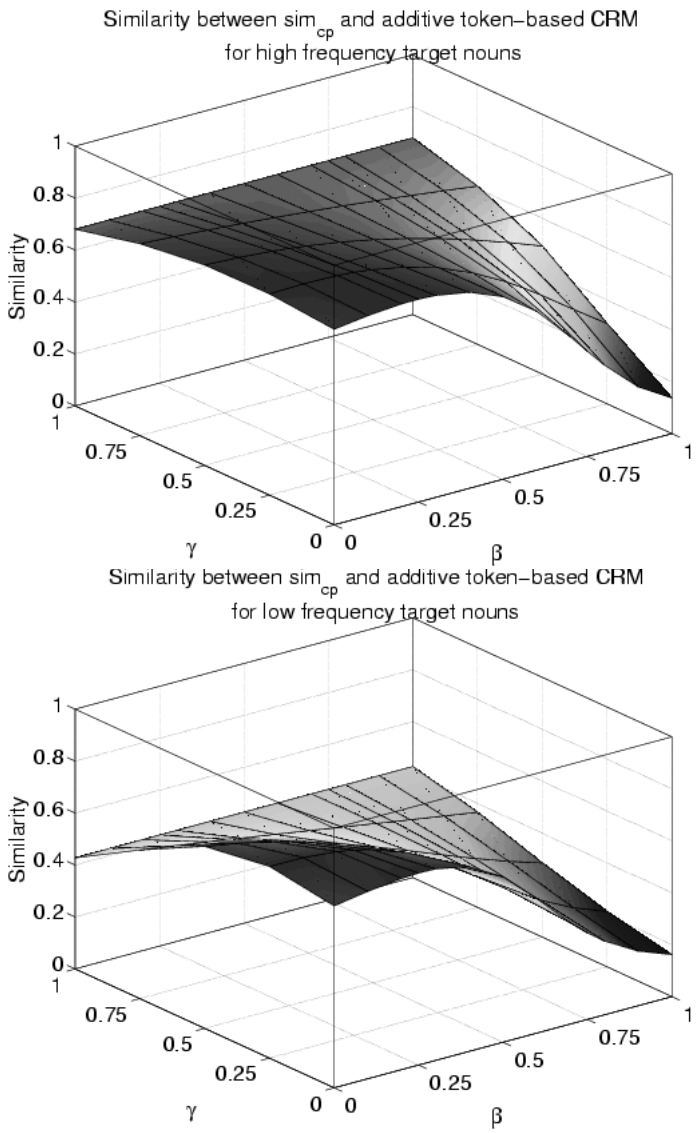

Figure 4.10: Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive token-based CRM and of sim_{cp}

Table 4.17 shows the optimal mean similarities between neighbour sets derived using each of the CRMs and $sim_{cp_{rev}}$. As usual, the mean similarities are calculated separately for the 1000 high frequency target nouns and the 1000 low frequency target nouns.

Although not very closely simulated by any of the CRMs, the reversed confusion probability is most closely simulated by the difference-weighted MI-based CRM. Figure 4.11 shows how the mean similarity between neighbour sets varies with β and γ for this measure and this CRM.

We might have expected that $sim_{cp_{rev}}$ would be a high precision measure, in contrast to sim_{cp} which is a high recall measure. However, the relationship between neighbour set similarity and parameter settings, observed in Figure 4.11, suggests that $sim_{cp_{rev}}$ neighbour sets contain a fairly even mix of high precision and high recall neighbours.

CRM	Target noun frequency					
	high			low		
	γ	β	Sim.	γ	β	Sim.
add-type	0.25	0.7	0.39 (0.12)	0.5	0.7	0.37 (0.11)
add-tok	0.25	0.8	0.30 (0.11)	0.5	0.9	0.25 (0.13)
add-mi	0.5	0.9	0.57 (0.12)	0.75	0.5	0.57 (0.11)
diff-type	0.25	0.9	0.28 (0.10)	0.25	0.25	0.37 (0.12)
diff-tok	-	-	0.39 (0.15)	-	-	0.43 (0.17)
diff-mi	0.0	0.9	**0.60 (0.12)**	0.5	0.3	**0.59 (0.12)**

Table 4.17: Mean optimal similarities between CRMs and $sim_{cp_{rev}}$. Figures in brackets are standard deviations.

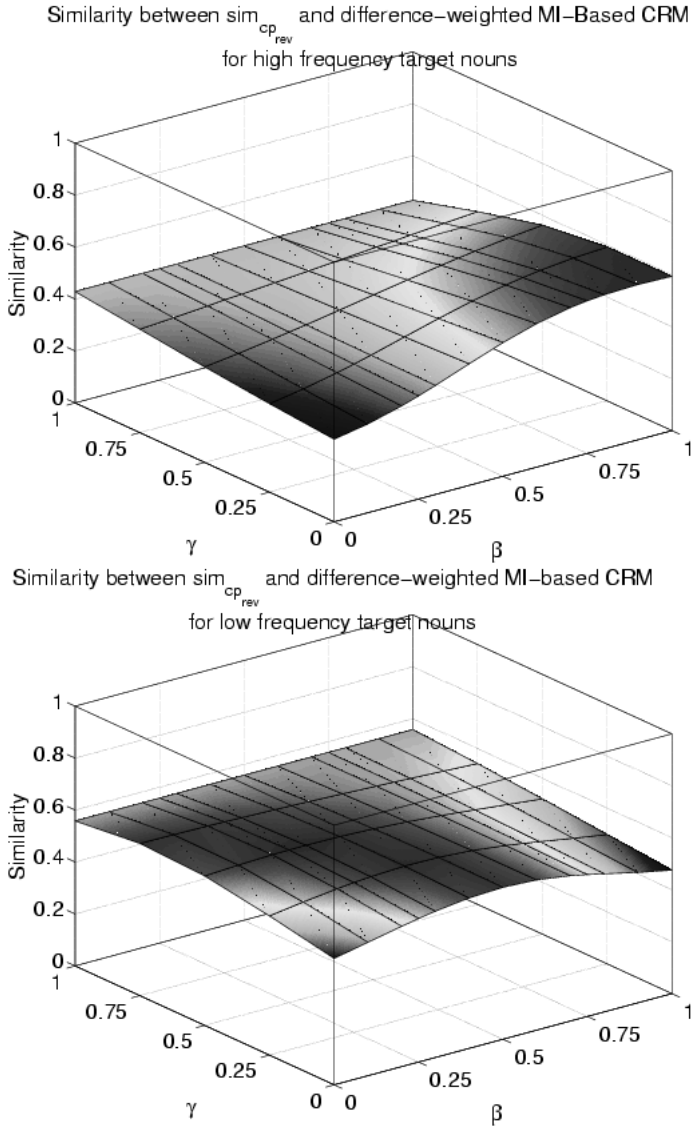

Figure 4.11: Variation (with parameters β and γ) in mean similarity between neighbour sets of the difference-weighted MI-based CRM and of $sim_{cp_{rev}}$

4.3.9 Hindle's Measure

Recall from Chapter 3 that the variant of Hindle's Measure considered here, sim_{hind+} is defined as:

$$sim_{hind+}(n_1,n_2) = \sum_{T(n_1)\cap T(n_2)} min(I(v,n_1),I(v,n_2)) \text{ where } T(n) = \{v : I(v,n) > 0\} \tag{4.62}$$

This expression is the same as the numerator in the expressions for precision and recall in the difference-weighted MI-based CRM:

$$\mathcal{P}_{mi}^{dw}(n_1,n_2) = \frac{\sum_{TP} I(n_1,v) \times \frac{min(I(n_1,v),I(n_2,v))}{I(n_1,v)}}{\sum_{F(n_1)} I(n_1,v)} = \frac{\sum_{TP} min(I(n_1,v),I(n_2,v))}{\sum_{F(n_1)} I(n_1,v)} \tag{4.63}$$

$$\mathcal{R}_{mi}^{dw}(n_1,n_2) = \frac{\sum_{TP} I(n_2,v) \times \frac{min(I(n_2,v),I(n_1,v))}{I(n_2,v)}}{\sum_{F(n_2)} I(n_2,v)} = \frac{\sum_{TP} min(I(n_2,v),I(n_1,v))}{\sum_{F(n_2)} I(n_2,v)} \tag{4.64}$$

since $TP = T(n_1) \cap T(n_2)$. However, we also note that the denominator in the expression for recall depends only on n_2 and therefore, for a given n_2 is a constant (similarly, the denominator of the expression for precision is a constant for a given n_1). Since n_2 is the target noun, it will remain the same as we calculate each neighbour set. Accordingly, the value of recall for each potential neighbour, n_1, of n_2 will be the value of sim_{hind+} divided by a constant. In other words, the value of $sim_{hind}(n_1,n_2)$ is directly proportional to recall in the difference-weighted MI-based CRM and we would expect the neighbour sets obtained by each to be identical. This can be shown empirically using the same technique as before. In Figure 4.12, mean similarity between neighbour sets is 1 at $(\gamma = 0, \beta = 0)$ for both high and low frequency nouns. As precision is taken into account by increasing either γ or β, similarity begins to drop. This is because the denominator in the expression for precision depends on the potential neighbour rather than the target noun.

4.3.10 Lin's MI-Based Measure

Recall from Chapter 3 that Lin's MI-Based measure is defined as:

$$\text{sim}_{\text{lin}}(n_1,n_2) = \frac{\sum_{T(n_1)\cap T(n_2)} (I(n_1,v) + I(n_2,v))}{\sum_{T(n_1)} I(n_1,v) + \sum_{T(n_2)} I(n_2,v)} \text{ where } T(n) = \{v : I(n,v) > 0\} \tag{4.65}$$

There are parallels between sim_{lin} and sim_{dice} in that both measures compute a ratio between what is shared by the descriptions of both nouns and the sum of the descriptions of each noun. The major difference appears to be the use of MI and hence we predicted that there would be a close relationship between sim_{lin} and the harmonic mean in the additive MI-based CRM. This relationship is shown below:

$$m_h(\mathcal{P}_{mi}^{add}(n_1,n_2), \mathcal{R}_{mi}^{add}(n_1,n_2)) = \frac{2.\mathcal{P}_{mi}^{add}(n_1,n_2).\mathcal{R}_{mi}^{add}(n_1,n_2)}{\mathcal{P}_{mi}^{add}(n_1,n_2) + \mathcal{R}_{mi}^{add}(n_1,n_2)} \tag{4.66}$$

$$= \frac{2.\frac{\sum_{TP} I(n_1,v)}{\sum_{F(n_1)} I(n_1,v)} . \frac{\sum_{TP} I(n_2,v)}{\sum_{F(n_2)} I(n_2,v)}}{\frac{\sum_{TP} I(n_1,v)}{\sum_{F(n_1)} I(n_1,v)} + \frac{\sum_{TP} I(n_2,v)}{\sum_{F(n_2)} I(n_2,v)}} \tag{4.67}$$

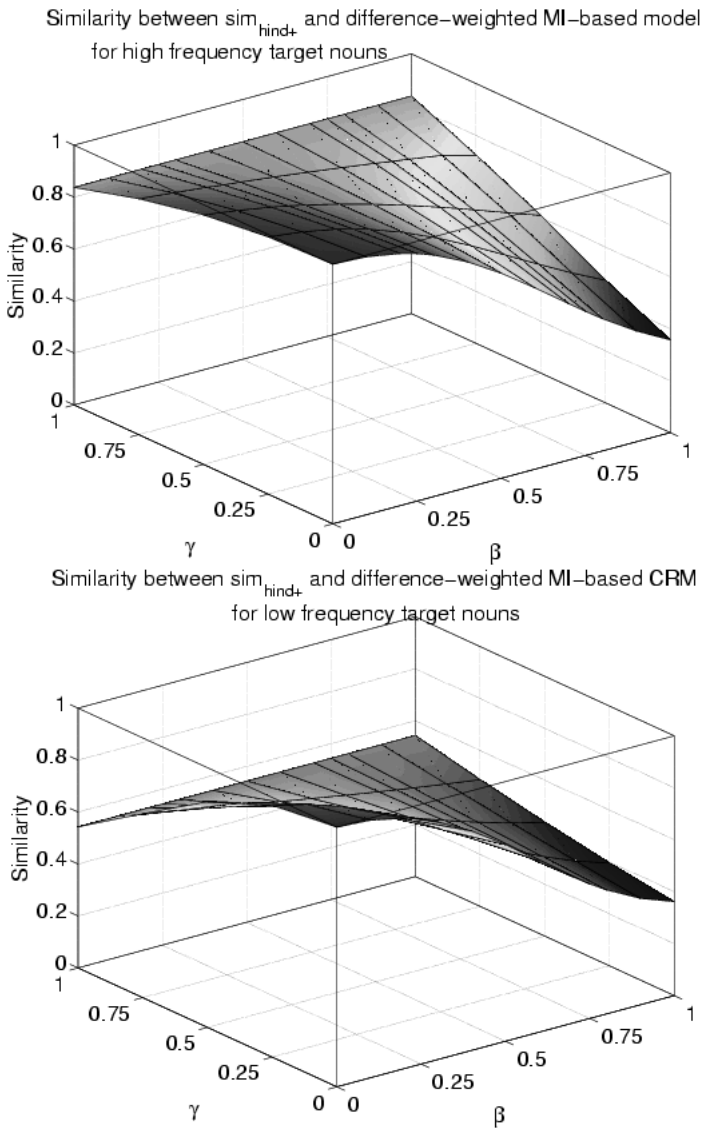

Figure 4.12: Variation (with parameters β and γ) in mean similarity between neighbour sets of the difference-weighted MI-based CRM and of sim_{hind+}

$$= \frac{2.\sum_{TP} I(n_1,v).\sum_{TP} I(n_2,v)}{\sum_{TP} I(n_2,v).\sum_{F(n_1)} I(n_1,v)+\sum_{TP} I(n_1,v).\sum_{F(n_2)} I(n_2,v)} \tag{4.68}$$

Now if $\sum_{TP} I(n_1,v) = \sum_{TP} I(n_2,v)$, it follows:

$$m_h(\mathcal{P}(n_1,n_2),\mathcal{R}(n_1,n_2)) = \frac{2.\sum_{TP} I(n_1,v)}{\sum_{F(n_1)} I(n_1,v)+\sum_{F(n_2)} I(n_2,v)} \tag{4.69}$$

$$= \frac{\sum_{TP} I(n_1,v)+I(n_2,v)}{\sum_{F(n_1)} I(n_1,v)+\sum_{F(n_2)} I(n_2,v)} \tag{4.70}$$

$$= \frac{\sum_{T(n_1)\cap T(n_2)} (I(n_1,v)+I(n_2,v))}{\sum_{T(n_1)} I(n_1,v)+\sum_{T(n_2)} I(n_2,v)} \text{since } T(n)=F(n) \tag{4.71}$$

$$= \text{sim}_{\text{lin}}(n_1,n_2) \tag{4.72}$$

Thus, when the additive MI-based model is used, $\gamma = 1$ and the condition $\sum_{TP} I(n_1,v) = \sum_{TP} I(n_2,v)$ holds, our framework reduces to sim_{lin}.

However, this necessary condition for equivalence is not one we can expect to hold for many (if any) pairs of words. Accordingly, in order to investigate how good an approximation the harmonic mean is to sim_{lin} in practice, we compared neighbour sets according to each measure using the neighbour set comparison technique outlined earlier.

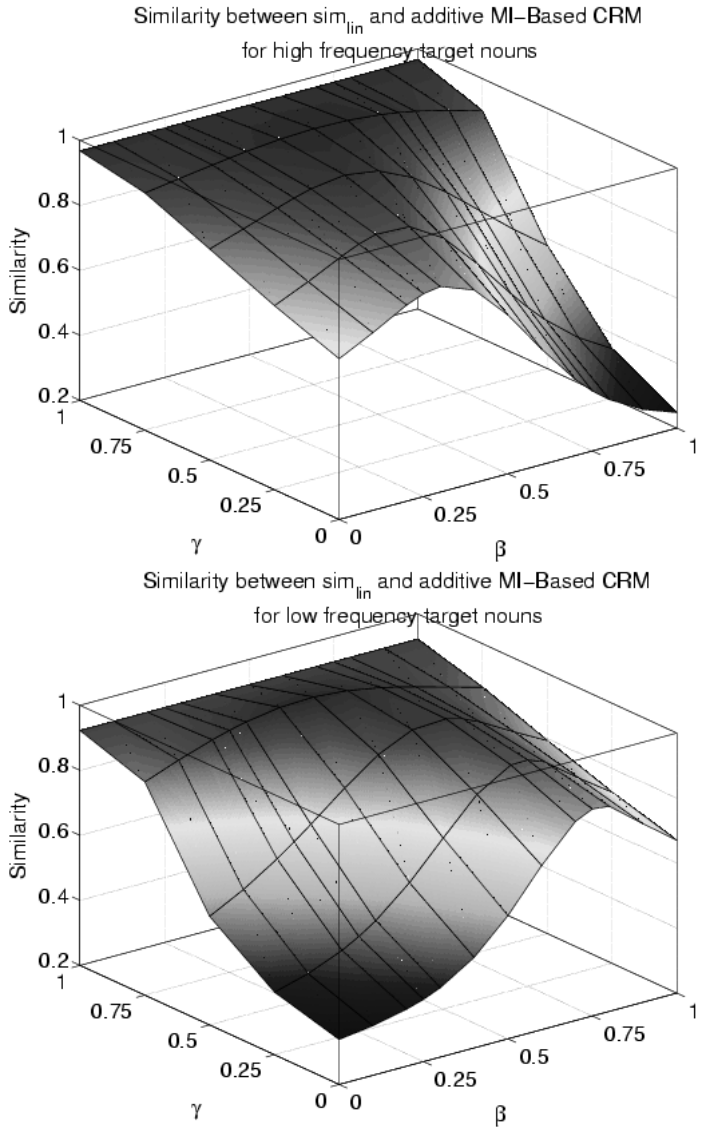

Figure 4.13: Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive MI-based CRM and of sim_{lin}

Figure 4.13 illustrates the similarity between sim_{lin} and the additive MI-based CRM for high and low frequency nouns. At $\gamma = 1$ the average similarity between neighbour rankings was 0.967 for high frequency nouns and 0.922 for low frequency nouns. This is significantly higher than similarities between other standard similarity measures. However the optimal approximation of sim_{lin} was found using $\gamma = 0.75$ and $\beta = 0.5$ in the additive MI-based CRM. With these settings, the average similarity was 0.987 for high frequency nouns and 0.977 for low frequency nouns (both of which differences are significant at the 99% level). This suggests that sim_{lin} allows more compensation for lack of recall by precision and vice versa than the harmonic mean.

4.3.11 MI Variant of Jaccard's Coefficient

We do not expect the $sim_{jacc_{mi}}$ to be exactly predicted by the additive MI-based CRM since there is a fundamental difference in the way each utilises MI. The MI-based variant of Jaccard's measure only uses MI to determine which co-occurrences are a noun's features whereas the MI-based CRM uses MI to determine the features of a noun and as weights in the similarity calculation. Theoretically speaking, $sim_{jacc_{mi}}$ appears to sit halfway between sim_{jacc} and sim_{lin}. Accordingly, we would expect it to be best simulated by a $\gamma = 1$ setting in either the additive type based CRM or the additive MI-based CRM. The optimal parameter settings and optimal similarities with this measure for each of the CRMs are shown in Table 4.18.

CRM	Target noun frequency					
	high			low		
	γ	β	Sim.	γ	β	Sim.
add-type	0.75	0.9	0.80 (0.07)	1.0	-	0.85 (0.06)
add-tok	1.0	-	0.50 (0.13)	0.0	1.0	0.21 (0.08)
add-mi	1.0	-	**0.91 (0.05)**	0.75	1.0	0.81 (0.05)
diff-type	0.75	0.6	0.80 (0.10)	0.75	0.4	**0.86 (0.05)**
diff-tok	-	-	0.61 (0.17)	-	-	0.46 (0.13)
diff-mi	0.75	0.0	0.88 (0.07)	0.75	1.0	0.78 (0.06)

Table 4.18: Mean optimal similarities between CRMs and $sim_{jacc_{mi}}$. Figures in brackets are standard deviations.

As predicted, $sim_{jacc_{mi}}$ can be closely approximated by the additive type-based model and the additive MI-based model. However, this $sim_{jacc_{mi}}$ appears to be significantly closer to the MI-based CRM for high frequency nouns and significantly closer to the additive type-based model for low frequency nouns. Further, the difference-weighted type-based and MI-based models also have parameter settings at which the approximation of $sim_{jacc_{mi}}$ is fairly close. In all cases, as predicted, the best simulations are obtained at higher values of γ. However, the best simulation is not always at $\gamma = 1$. The variation in performance with β and γ for the additive MI-based CRM is illustrated in Figure 4.14. For low frequency nouns, performance is less at $\gamma = 1$ than ($\gamma = 0.75, \beta = 1$) but the difference observed (0.004) is statistically insignificant at any level.

4.4 Conclusion

We have defined a framework for measuring lexical distributional similarity based on the concept of substitutability. We have cast the problem as one of *co-occurrence retrieval* for which we can measure precision and recall by analogy with the way that they are measured in *document retrieval*.

Since similarity is defined by substitutability, it is inherently asymmetric. However, the CR framework does not require similarity to be viewed as asymmetric since there are settings in each model (($\gamma = 1, \beta$) and ($\gamma, \beta = 0.5$)) at which it is symmetric and there is a model (the difference-weighted token-based model) which is always symmetric. These symmetric settings may be thought of as capturing our notion of *inter-substitutability*. The potential for exploiting the asymmetry in similarity will be explored in future chapters in the context of applications of distribu-

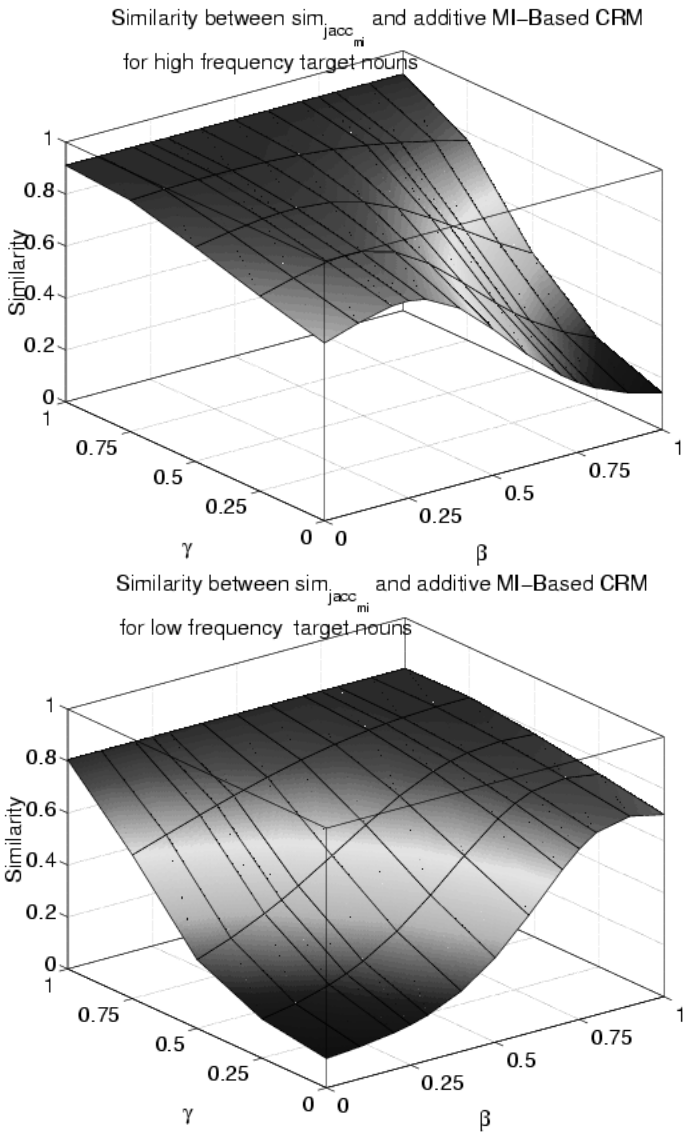

Figure 4.14: Variation (with parameters β and γ) in mean similarity between neighbour sets of the additive MI-based CRM and $sim_{jacc_{mi}}$

tional similarity.

This idea of asymmetry in substitutability makes our framework comparable with measures based on relative entropy or the confusion probability. Since $\mathcal{R}(n_1,n_2) = \mathcal{P}(n_2,n_1)$, we could consider using $D(n_1||n_2)$ and $D(n_2||n_1)$ instead of precision and recall. However, our framework has the advantages that it directly defines a similarity measure, does not require smoothing of the base language model and allows us to explore systematically other questions we have about the relative importance of different features. In order to do this, we have defined six different co-occurrence retrieval models (CRMs). Three of the models are additive models and the other three are difference-weighted models. This allows us to explore the effect of the difference in feature values for shared features. For both additive and difference-weighted models, we have defined type-based, token-based and MI-based models. This allows us to explore the effect of weighting different features by some degree of relevance or association.

The rest of this chapter has been devoted to analysing the neighbour sets produced by the CR

framework and determining the CR characteristics of existing distributional similarity measures. We saw that, in general, high recall neighbours tend to be high frequency nouns, high precision neighbours tend to be low frequency nouns and a high harmonic mean tends to produce neighbours of a similar frequency to the original noun. However, the CRM used also has an effect on the frequency of the neighbours produced. In particular, the difference-weighted models tend to be flatter with respect to the parameters β and γ and produce neighbours of a similar frequency to the target noun. Also of note is the $(\gamma = 0, \beta = 0.5)$ setting in the additive MI-based model which appears to produce neighbour sets with no particular tendencies to high or low frequency nouns.

With regard to existing distributional similarity measures, we have seen that six of those considered can be directly derived or very closely approximated by the CR framework. A summary of these findings was presented in Table 4.9. Knowledge of the CR characteristics of existing measures helps us to understand these measures. For example, high recall measures such as sim_{τ}, sim_{hind}, $dist_{\alpha 1}$ and sim_{cp} tend to select higher frequency nouns. The measures that require less recall and more precision, $dist_{\alpha 2}$ and $sim_{cp_{rev}}$, will select lower frequency neighbours than the high recall measures. The measures which require both high recall and high precision, sim_{jacc}, sim_{dice}, sim_{lin} and $sim_{jacc_{mi}}$ select neighbours of a similar frequency to the target noun.

The CR framework, however, more than simulates existing measures of distributional similarity. It defines a space of distributional similarity measures that is already populated with a few named measures. By exploring the space we can discover what are the desirable characteristics of distributional similarity measures. It may turn out the most useful measure within this space has already been discovered or it may be that we discover a new optimal combination of characteristics. The primary goal, however, is to understand how different characteristics relate to high performance in different applications and thus explain why one measure performs better than another.

With this goal in mind, we now turn to the applications of distributional similarity. In the next chapter, we consider what characteristics of distributional similarity measures are desirable in the application of language modelling. We will do this by considering how distributional similarity information can be used in language modelling and then by considering the performance of different models and different parameter settings within a pseudo-disambiguation task.

Chapter 5

Language Modelling

As discussed in Chapter 2 (Section 2.2.2), smoothing probability estimates for co-occurrences using nearest neighbour techniques is an intuitively plausible way of providing more reliable probability estimates. In this chapter we consider the evaluation of using such techniques in this application area. In Section 5.1 we discuss using nearest neighbours to derive smoothed co-occurrence probability distributions. We evaluate these probability distributions by considering their similarity to unseen empirical distributions. In Section 5.2 we compare a number of similarity measures, including the CR framework presented in Chapter 4, for use in this area using pseudo-disambiguation experiments.

5.1 Deriving Probability Distributions

In this section we consider the use of nearest neighbours to derive smoothed co-occurrence probability distributions. We will first consider the intuitions behind nearest neighbour smoothing and the aims of this section. We will then present our smoothing algorithm, our method of evaluation and our results.

5.1.1 Introduction

If a very low probability verb, v, which is equally plausible with two nouns, n_1 and n_2, has only been observed once in a sample of language data, probability theory tells us that it should have occurred 0.5 times with each noun whilst in reality we know that, due to the discretization of data, it must have occurred a single time with just one of the nouns. As we see more data, however, we would expect that the probabilities observed in the data should approach the theoretical population probabilities and therefore we should see the verb occurring a roughly equal number of times with each noun.

The aim in smoothing is to provide better probability estimates than the sample distribution and, thus, to make it possible to predict what co-occurrences might be observed as more data is seen. Smoothing algorithms can achieve an increase in performance over Maximum Likelihood Estimation MLE by simply assuming that nothing is impossible. Add1 smoothing assumes that all co-occurrences have an equal prior probability of occurring and back-off smoothing assumes that,

in the absence of reliable empirical estimates, events, e.g. nouns and verbs, occur independently. The assumption in nearest-neighbour smoothing, however, is that words that have been seen to be similar in terms of their co-occurrences are likely to share other co-occurrences that have not been observed. This is the basis of a body of work on nearest-neighbour smoothing (Hindle, 1990; Brown et al., 1992; Dagan et al., 1993; Pereira et al., 1993; Dagan et al., 1999).

There are two hypotheses that we aim to investigate in this section. The first is that nearest neighbour smoothing is a plausible approach to smoothing. In the above scenario, the two nouns, n_1 and n_2, which should share the verb co-occurrence v are likely to have been seen to share other verb co-occurrences. Accordingly, a distribution that has been smoothed using similar distributions should be more similar to the population distribution than the original sample distribution. Consequently, it should also be more similar than the original random sample to another random sample taken from the same population.

The second hypothesis is that sample distributions for more frequently occurring nouns are closer to the population distributions than those for less frequently occurring nouns. If verb v can occur occur with n_1 and n_2, then a single co-occurrence is most likely to be with which noun is overall most frequently occurring. More generally, we would expect that most of the changes required to the distribution of a very high frequency noun will be small probability adjustments to allow probability mass to be reassigned to lower frequency nouns. The changes to the distributions of lower frequeny nouns will include the addition of new features and changes to probabilities of a single feature will be more significant with respect to the whole distribution, since a lower frequency noun will initially have less features.

In order to investigate the first of these hypotheses, we smooth empirical distributions for the 1000 most frequent nouns using their distributionally nearest neighbours and measure the distance for each from the new distribution to the original distribution and to an unseen empirical distribution. We expect to see that the derived distribution is closer than the original distribution to the unseen distribution. Ideally, the derived distribution should be at the same minimal distance from both empirical distributions, since the population distribution would predict each with equal error.

In order to investigate the second hypothesis, we use an iterative smoothing algorithm which stops when either a certain number of neighbours have been considered or when a certain amount of change to the original distribution has been caused. We expect to see that determining the amount of change allowed as a function of word frequency will lead to improved results over using a fixed number of neighbours for each word.

However, since both the smoothing algorithm and the evaluation technique rely on a distributional similarity measure (one to determine neighbours and the distance from the original distribution and the other to determine distance of the resultant distribution from empirical distributions), we do not compare distributional similarity measures using this approach. For example, a bad similarity measure could base its similarity estimate on the most frequently occurring feature. We might not expect this measure to be very good at finding neighbours but it could find that the derived distributions were very close to the empirical distributions. We do not expect any of the distributional similarity measures considered here to be this bad, but it does still seem circular to evaluate distributional similarity measures using distributional similarity measures. Con-

sequently, we only attempt to evaluate the two hypotheses above using this approach and use pseudo-disambiguation experiments to compare measures.

The similarity measure chosen for this work is the α-skew divergence measure because relative entropy, on which it is based, is an established way of measuring the cost of using a model distribution in place of an empirical distribution in language modelling. However, it may be that a different distributional similarity measure should be used both in both the smoothing process and in the evaluation process. This is something that we would like to explore in the future.

Finally, we note that the plausibility of nearest-neighbour smoothing was demonstrated by Dagan et al. (1999), using experiments on which our own are based. However, we feel that it is worth us redemonstrating it here for three reasons. First, the plausibility of nearest-neighbour smoothing is an important issue, since it motivates a substantial amount of the work on distributional similarity measures. Second, our work differs from that of Dagan et al. (1999) in our secondary hypothesis, which it would be difficult to show in the context of nearest neighbour smoothing without first demonstrating the plausibility of nearest neighbour smoothing. Third, we will use these experiments as a basis for cross-validating pseudo-disambiguation as a method for comparing distributional similarity measures in Section 5.2.

We will now turn our attention to the smoothing algorithm used.

5.1.2 Smoothing Algorithm

The original data-set for these experiments consisted of the 2,852,300 lemmatised noun-verb direct-object pairs described in Chapter 3. We selected the data pertaining to the 1000 most frequent nouns and divided it into two halves: one for training and the other for testing. The distributional similarity is calculated between each pair of nouns using the training data and an ordered list of neighbours constructed for each noun.

The smoothed, or derived, distribution for each of the 1000 target nouns is initialised with the sample distribution for that noun i.e., a vector of verbs and their associated conditional probabilities ($P(v|n)$) as observed in the training data. A noun's current distribution, C_i, is *updated* with the distribution of its next nearest neighbour N_i by calculating a weighted average of the two distributions. This process is repeated for each next nearest neighbour until a *stopping condition* is met. This process is illustrated in Figure 5.1. We will now discuss the calculation of weights and three possible stopping conditions.

5.1.2.1 Combining Probability Distributions

A simple way to combine two distributions is to take the average of the corresponding probabilities in each distribution. This approach, however, does not take into consideration either the rank of the neighbour or its similarity to the target noun. Intuitively, if the neighbour is identical to the target noun, we cannot learn any new noun-verb co-occurrences from that neighbour. However, we would not expect very different population distributions to give rise to identical sample distributions and therefore we can become more confident that the sample distributions are close to the population distributions. As neighbours become less similar to the target noun, they will add more new noun-verb co-occurrences to the distribution of the target noun, but we can be less confident that these verbs would actually co-occur with the target noun. Hence, these co-occurrences should be assigned lower probabilities. However, if many neighbours suggest the

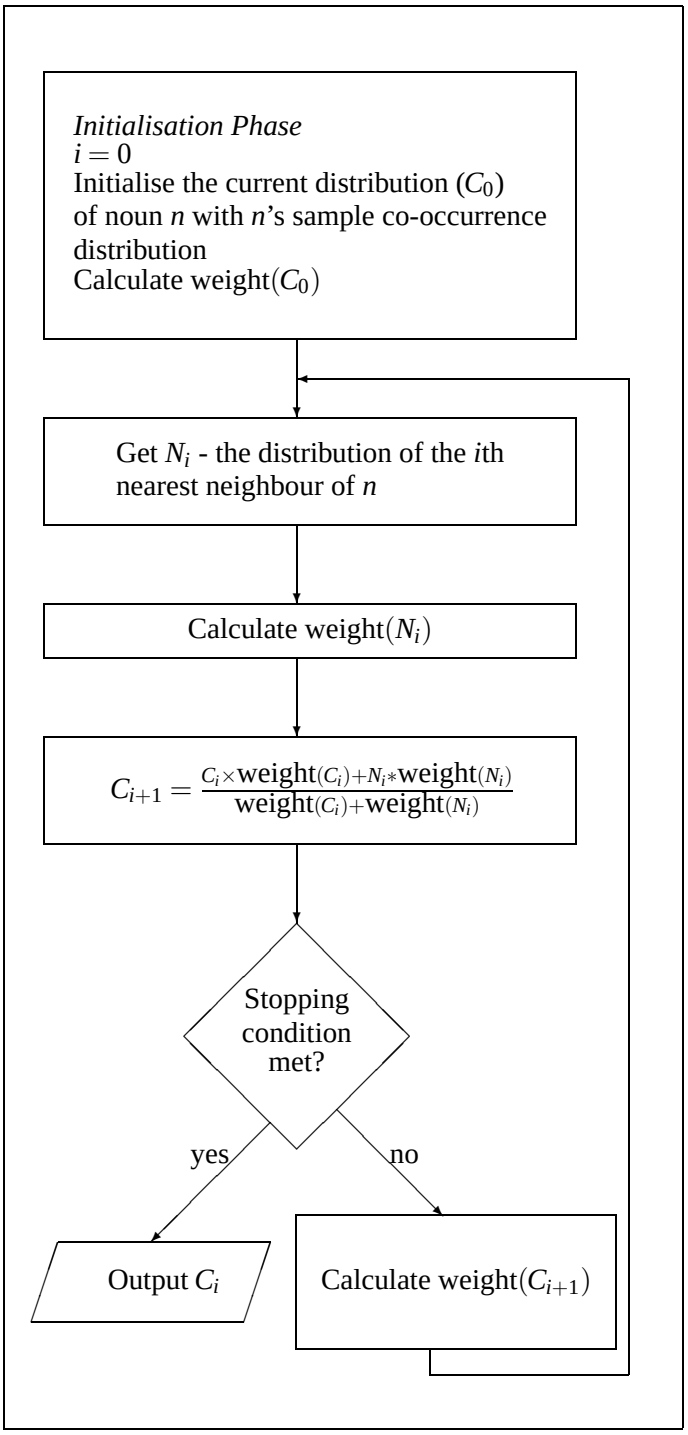

Figure 5.1: Algorithm for smoothing co-occurrence distribution of target word using nearest neighbours

same verb co-occurrence, our confidence that this verb can co-occur with the target noun increases and the probability assigned to the co-occurrence should increase as a consequence.

These intuitions are modelled by assigning weights to both the current distribution and the neighbour distribution and then calculating a weighted average of the two distributions.

The weight of a neighbour's distribution depends on its similarity to the target noun's distribu-

tion. When the similarity function is $dist_{\alpha 1}$, the neighbour distribution weight is calculated using a negative exponential function of that neighbour's distance from the target noun. If n_1 is the target noun and n_2 is the neighbour under consideration, the weight given to n_2's distribution is given by:

$$e^{-dist_{\alpha 1}(n_2,n_1)} \tag{5.1}$$

This function is a special case of the function proposed by Dagan et al. (1999) (where $\beta = 1$) and can be justified as follows. The similarity of two nouns is calculated as the average log ratio of the corresponding probabilities in each distribution. Accordingly, we use the inverse of this to compute the weight given to a probability estimate from the neighbour's distribution.

Since a word has a distance of 0 from itself (when using a distance function such as $dist_{\alpha 1}$), the weight given to the initial *current* distribution is 1. However, once the weighted average has been calculated, the new distribution, which becomes the current distribution in the next iteration, is given a weight which is the sum of the weights of the two individual distributions — thus simulating increasing confidence in the current distribution as more data is seen.

5.1.2.2 Stopping Conditions

We output the current distribution as the new distribution, once a certain condition has been met. In order to investigate our secondary hypothesis, we have experimented with three different stopping conditions:

1. Stop once a fixed value of k (the number of nearest neighbours) has been exceeded.
2. Stop if combining the next neighbour would mean that the distance between the original distribution and the new distribution exceeds a fixed threshold.
3. Stop if combining the next neighbour would mean that the distance between the original distribution and the new distribution exceeds a threshold that is determined as a function of the frequency of the target noun.

Using the first condition, the same number of neighbours is considered for each target noun. The number considered, k, is a global parameter, which is optimised over all target nouns. The second stopping condition allows the number of neighbours considered to vary from word to word. We stop when combining the next neighbour would cause too much change to the original empirical distribution. Since the α-skew divergence measure determines how well one distribution approximates another, we measure the amount of change using $dist_{\alpha 1}(C_0, C_{i+1})$, where C_0 is the original unsmoothed distribution and C_{i+1} is the proposed new distribution. Note that this is different to only considering neighbours within a fixed similarity distance of the target noun since the exact cut-off point will depend on the changes caused by nearer neighbours. A neighbour that is similar to other neighbours, and thus is reinforcing changes made by these neighbours, is more likely to be allowed than a neighbour which is dissimilar to other neighbours.

Using the third stopping condition we simulate having more confidence in maximum likelihood estimates pertaining to high frequency target nouns. In other words, the distributions of high frequency target nouns are allowed to change less than the distributions of low frequency target nouns.

The intuition behind the third stopping condition is strongly supported by an initial statistical study. In this study, the distance between one half of the data set (the training data) and the whole

data set, which is the best estimate we have of population probabilities, was measured for each noun using $dist_{\alpha 1}$. This distance is plotted against word frequency in Figure 5.2.

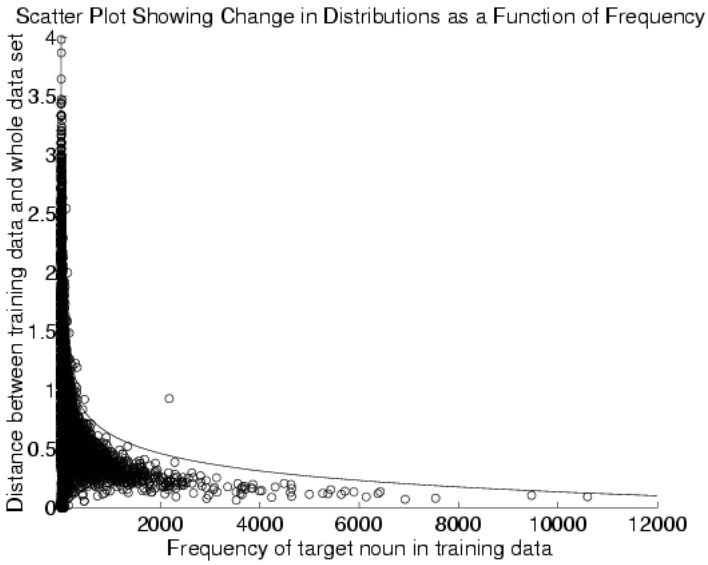

Figure 5.2: Distance between half the data set and the whole data set plotted against word frequency

Ignoring one anomalous point[1], there appears to be a smooth line which puts an upper bound on the distance from half the data set to the whole data set. This line can be fitted by a function of the form:

$$\text{Threshold}(n_1) = 2.3 \times \kappa - \kappa \times \log(\log(freq(n_1))) \tag{5.2}$$

We use this function to implement the third stopping condition, leaving κ as a parameter to be optimised. The shape of the function, with $\kappa = 1.7$, is shown in Figure 5.2.

5.1.3 Evaluation

Central to the evaluation task is the idea that, in language modelling tasks, we have a *seen* chunk of co-occurrence data and we want to predict the probabilities of the co-occurrences in the next or an *unseen* chunk of co-occurrence data. Maximum likelihood estimation predicts that the unseen chunk is identical to the seen chunk. The aim of a smoothing algorithm is to make changes to the distributions in the seen chunk in order to better predict the unseen chunk. Of course, no sample is perfectly representative of the population but our estimated population probabilities should ideally be able to predict any random chunk with approximately the same error.

Accordingly, we measure the distance from potential model distributions to each half of the data set using the α-skew divergence measure. This technique is akin to using relative entropy to measure the cost of using the model distribution in place of the empirical distribution. Using the α-skew divergence measure instead of relative entropy means that we do not penalise as heavily a model distribution that cannot provide an estimate for every co-occurrence in the empirical

[1]This anomalous point corresponds to a word for which there was much less than the expected 50% of the total data in the training data (approximately 20%)

distribution. This approach is approximately equivalent to implementing a hierarchical smoothing algorithm which backs-off to unigram probabilities in the absence of similarity-based probability estimates and then evaluating using relative entropy.

In this paradigm, an optimal smoothing technique would provide estimated probability distributions at the same minimal distance from the seen (i.e. training data) distributions and the unseen (i.e. test data) distributions. Using the same paradigm, we also evaluate two baselines.

5.1.3.1 Baseline 1: MLE

MLE predicts that the population probabilities are exactly the same as the sample probabilities. Accordingly, it perfectly predicts the training data (i.e. is at zero distance from it) but not the test data. Performance of MLE is therefore calculated as the distance between the training data and the test data i.e. the ability of the training data to predict the test data.

5.1.3.2 Baseline 2: AddX for the N Most Frequent Verbs

Coverage of the test data (i.e. the number of co-occurrences in the test data for which a probability can be calculated) using similarity-based smoothing techniques is undoubtedly going to be higher than when using MLE. The reason for this is that any verb that occurs with any neighbour (before the stopping condition is met) will be added with some small probability to the distribution of the target noun. In the limit when all neighbours are considered, all verbs that occur in the training data will appear in all noun distributions. Before that limit is reached it seems likely that more frequent verbs will be added more frequently or with a higher probability, since they tend to occur with a greater number of nouns.

Accordingly, we also consider the baseline where a fraction (X) of a co-occurrence is added for each of the N most frequent verbs. X and N are parameters which are optimised, making it a generalisation of Add1 smoothing ($X = 1$ and $N = \infty$).

5.1.4 Results

For each probability estimation technique (and stopping condition and parameter setting where applicable) we computed the mean similarity of the estimated distributions to the training data distributions, the mean similarity of the estimated distributions to the test data distributions and the coverage (type and token) of the test data by the estimated distributions. The results for each technique are summarized in Table 5.1.

The coverage results for MLE show that, without smoothing, almost 50% of distinct co-occurrence types in the test data are not observed in the training data. The coverage for tokens is of course a lot higher, since those co-occurrences types which do not occur in the training data tend to each occur only once in the test data. However, with only maximum likelihood techniques, we would incorrectly conclude that 14% of the unseen data was impossible.

AddX for the N most frequent verbs is a naïve smoothing algorithm which enables us to increase the coverage of the test data. Maximum coverage can be achieved by adding occurrences between every noun and verb seen in the training data. Note that maximum coverage does not equal 100% unless allowance is made for verbs previously unseen with any noun. Ignoring these unseen verbs, maximum co-occurrence type coverage of the unseen data set is 98.5% and maximum co-occurrence token coverage is 99.6%. However, we found that maximising coverage was at the expense of decreasing similarity with the unseen data. This is because other co-occurrence

	optimum parameter setting	similarity to training data		similarity to test data		coverage	
		mean	σ	mean	σ	type	token
Stopping condition 1: (k nearest neighbours)	k=5	0.2501	0.0348	0.8207	0.2665	85.7%	96.4%
Stopping condition 2: (max change: fixed threshold (t))	t=0.7	0.2382	0.0341	0.8278	0.2686	86.6%	96.6%
Stopping condition 3: (max change: threshold = f(freq,κ))	κ=1.7	0.2376	0.0579	**0.8149**	0.2703	83.6%	95.7%
Baseline 1: (Maximum likelihood estimation)	n/a	0	0	1.037	0.3940	52.2%	86.0%
Baseline 2: (AddX for N most frequent verbs)	N=700 X=0.15	0.1244	0.0691	0.8709	0.3214	79.3%	94.5%

Table 5.1: Comparison of stopping conditions and baselines for language modelling

types, not seen in the test data, are also added. Maximum similarity with the unseen data was found at N=700, X=0.15. In other words, increasing N beyond 700 leads to smoothed distributions with significantly lower similarity to both the training and test data. This remains true even if X is decreased to compensate for the increase in previously unseen events now assigned a probability.

All of the nearest neighbour techniques significantly outperform the baselines in terms of similarity to the test data[2] and in terms of coverage. This suggests that they are more discriminating in their selection of verbs to be added to the target noun's distribution. The best results for this data set are achieved using the third stopping condition. However, we note that the differences between the three stopping conditions are not statistically significant.

The variation with the parameter κ for the third stopping condition is illustrated in Figure 5.3. The left-hand graph shows the variation in similarity with the test and training data with κ and the right-hand graph shows the variation in coverage with κ. From Figure 5.3 we can see that the optimal value of κ is around 1.7. This value corresponds to the function illustrated in figure 5.2. However, normally, it would not be possible to fit an upper bound in advance (since it relies on knowing the unseen data that will be observed).

Finally, it is difficult to know how well, in absolute terms, we could hope to do on this task. We can compute a lower bound on absolute performance using the whole data set, since this is the best estimate of population probabilities that we have. The whole data set has a mean similarity with the training data of 0.158 (standard deviation = 0.0613) and a mean similarity with the test data of 0.187 (standard deviation = 0.0986). However, this lower than the real lower bound. In reality, the population distribution would not be this close to either half of the data-set, because it will contain information about possible co-occurrences not seen in the whole of our data set. However, the smoothed distributions all predict the training data much better than they predict the test data. This suggests that there is still considerable room for improvement in estimating population probabilities.

[2] All of the differences are statistically significant at the 99% level — i.e. there is a less than 1% chance of these differences occurring by chance.

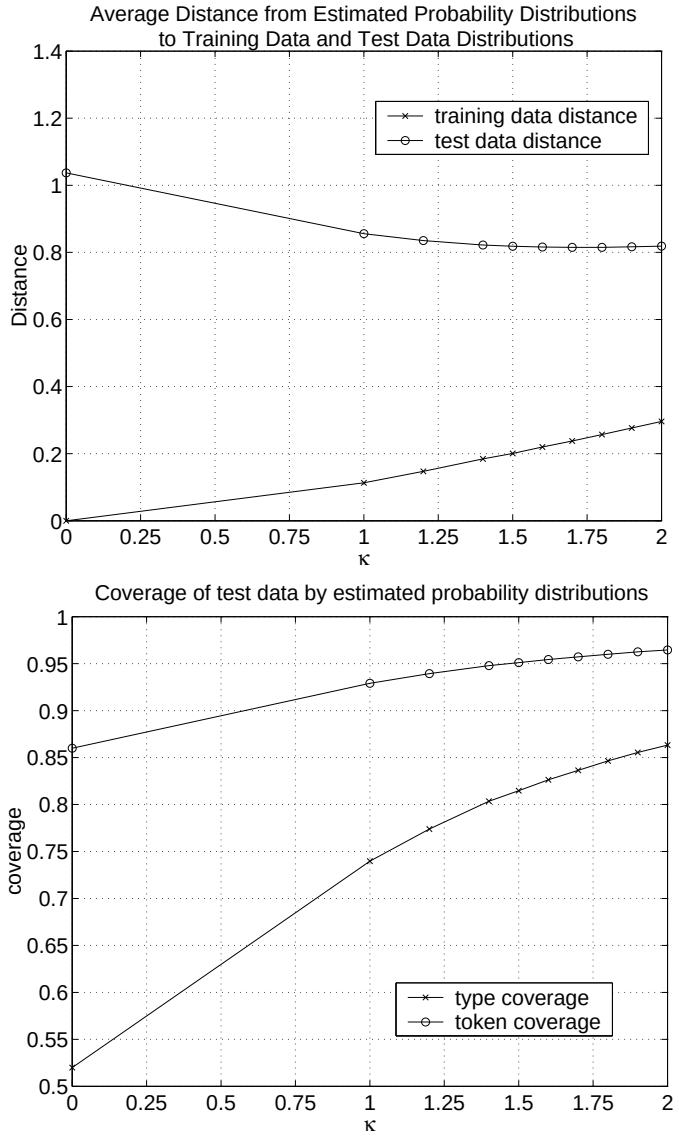

Figure 5.3: Results for stopping condition 3 in the language modelling task

There are a number of possible reasons for these apparently poor results. The first is that by only considering 1000 nouns, we are limiting the neighbours selected and the co-occurrences which can be added to the distributions. However, recall from Chapter 3, $dist_{\alpha 1}$ tends to select very high frequency neighbours even if there are other lower frequency potential neighbours. Accordingly, considering more potential neighbours would not change the 1000 nearest neighbours of the 1000 most frequent nouns. Having to consider more than 1000 nearest neighbours would not only be computationally expensive but would also make us doubt that it was the effect of the "nearest" neighbours that was leading to improved smoothing results.

A second possible reason is that we are not using optimal weight functions in our smoothing algorithm. Dagan et al. (1999) explored weight functions for nearest neighbours of the form:

$$e^{\beta dist_{\alpha}(n_2,n_1)} \tag{5.3}$$

where β is used to control the influence of more distant neighbours. We have used $\beta = 1$ in

our experiments whereas Dagan et al. (1999) show that increasing β (up to a value of 4) leads to improved performance. In the context of our experiments, increasing β would result in more neighbours being considered for the same amount of change to the original distribution and therefore would undoubtedly improve coverage. In order to investigate whether it would also improve the similarity with the unseen data set we repeated our experiments for the first and third stopping conditions using a value of $\beta = 3$. Using the first stopping condition and $\beta = 3$, we were able to obtain optimal similarity to the test data of 0.825 ($\sigma = 0.321$) at $k = 300$. This corresponded to a much smaller average change to the training data (average similarity = 0.115) but did not get us any closer to the test data. Using stopping condition 3, we were able to obtain optimal similarity to the test data of 0.8140 ($\sigma = 0.322$) at $\kappa = 1.0$. This again corresponded to a much smaller average change to the training data (average similarity = 0.108) and only gets us minimally (and statistically insignificantly) closer to the test data. These results would suggest that the value of β has less effect in our smoothing algorithm than in that of Dagan et al. (1999). This is likely to be because we stop considering more neighbours when they become too distant from the original distribution and because we increase the weight of the current distribution with each iteration, which effectively reduces the impact of more distant neighbours.

A third possible reason is that we are using the wrong distributional similarity measure to determine the nearest neighbours of a target noun. We will investigate this in Section 5.2 using pseudo-disambiguation experiments.

5.1.5 Conclusion

We have demonstrated the plausibility of similarity-based smoothing in language modelling. We have shown that such techniques are more discriminating in their selection of verbs to be added to the target noun's distribution than naïve Add1 type smoothing. Further, the probabilities estimated allow different, previously unseen, co-occurrences to be distinguished probabilistically. The other major approach to smoothing that allows us to do this is back-off (Katz, 1987). However, this leads to the slightly counter-intuitive result that unseen co-occurrences of high probability events are deemed more probable than unseen co-occurrences of low probability events. In the future, we would like to compare results for nearest-neighbour smoothing and back-off smoothing. However the goal of this work is not to carry out smoothing but to determine desirable characteristics of a distributional similarity measure which might be used in nearest-neighbour smoothing. Hence, it is sufficient for us to be confident that the nearest neighbours approach is a plausible approach to smoothing.

We have also shown that determining the amount of change allowed to the original unsmoothed distribution as a function of word frequency leads to small improvements in performance.

However, our main interest is in comparing similarity (or distance) measures. As already discussed, the approach used here is not really appropriate for this since the evaluation methodology also relies on a distributional similarity measure. In the next section we turn to pseudo-disambiguation experiments as a method of evaluating different similarity measures within the language modelling environment. For efficiency reasons and to avoid the issues of choosing an appropriate weight function we follow Lee (1999) and use a nearest neighbour voting scheme rather than smoothed probability distributions. However, there is no reason to suspect that the

effects observed would not carry over into the smoothing algorithm. Further, we shall show at the end of Section 5.2 that this is indeed the case by reimplementing these smoothing experiments using the best-performing similarity measure.

5.2 Pseudo-Disambiguation Experiments

As discussed in Chapter 2 (Section 2.3.5), pseudo-disambiguation tasks have become a standard evaluation technique. In the current setting, we may use a word's neighbours to decide which of two co-occurrences (one of which occurred in a portion of held-out test data and one of which did not) is more likely. As already discussed, this task has relevance in language modelling and word sense disambiguation applications. However, there are various different approaches to how such a task might be set up and this will now be discussed. Having discussed our methodology, we will present results on the pseudo-disambiguation task for a number of different similarity measures.

5.2.1 Experimental Set-up

In this section, we discuss the set-up of pseudo-disambiguation experiments. We start by describing our own approach. We then compare this with the approach of Lee (1999) and discuss a number of statistical biases that we observed when using this approach. Our approach is intended to minimise these biases.

The data-set used in these experiments is the same as that described in Chapter 3 (Section 3.2.1). From the starting point of 2,852,300 lemmatised noun-verb direct-object pairs extracted from the BNC using the RASP parser, we first discarded all co-occurrence pairs where the noun does not occur in WordNet (since we also consider a WordNet-based similarity measure). We then selected all of the data pertaining to a set of target nouns. Since we are interested in the effects of word frequency on similarity, we consider 1000 high frequency target nouns and 1000 low frequency target nouns. The high frequency nouns are the 1000 most frequently occurring nouns in our direct-object data, which corresponds to a frequency range of [586, 20871]. The low frequency nouns were selected as frequency ranks 3001 to 4000 in the data, which corresponds to a frequency range of [72,121]. The nouns and their associated frequencies used in these experiments are listed in Appendix A. After discarding data not pertaining to the 2000 target nouns, the total amount of data available was 1,995,998 co-occurrence tokens.

For each target noun, 80% of the available data was randomly selected as training data and the other 20% set aside as test data. The training data is used to compute similarity scores between all possible pairwise combinations of the 2000 nouns and to provide (MLE) estimates of noun-verb co-occurrence probabilities in the evaluation task. The test data provides unseen co-occurrences for evaluation purposes. We converted each noun-verb pair (n, v_1) in the test data into a noun-verb-verb triple (n, v_1, v_2). v_2 is chosen such that $P(v_2)$ is approximately equal to $P(v_1)$ over all the training data. This is done by constructing a ranked frequency list of verbs and their associated frequencies. We randomly select v_2 from the verbs that have the same frequency as v_1 plus or minus 1. If there are no other verbs within this frequency range then the test instance is discarded. This method ensures that there are no systematic bias towards v_2 being of a higher or lower frequency than v_1. We also ensure that (n, v_2) has not been seen in the test or training data. Ten test

instances[3] were then selected for each target noun in a two-step process of 1) whilst more than ten triples remained, discarding duplicate triples and 2) randomly selecting ten triples from those remaining after step 1. At this point we have 10,000 test instances pertaining to high frequency nouns and 10,000 test instances pertaining to low frequency nouns. Each of these sets were split into five disjoint subsets — each containing two instances for each target noun. We use these five subsets in two ways. First, we can perform five-fold cross validation. In five-fold cross validation, we compute the optimal parameter settings in four of the subsets and the error rate at this optimal parameter setting in the remaining subset. This is repeated five times with a different subset held out each time. We can then compute an average optimal error rate. We cannot, however, compute an average optimal parameter setting since this would assume a convex relationship between parameter settings and error rate. In order to study the relationship between parameter settings and error rate, we combine three of the sets to form a development set and two of the sets to form a test set. The development set is used to optimise parameters and the test set to determine error rates at the optimal settings. In graphs showing the relationship between error rate and parameter settings, it is the error rate in this development set that is shown.

Having constructed the test sets, the task is to take each test instance (n, v_1, v_2) and use the nearest neighbours of noun n (as computed from the training data) to decide which of (n, v_1) and (n, v_2) was the original co-occurrence. Each of n's neighbours, m, is given a vote which is equal to the difference in frequencies of the co-occurrences (m, v_1) and (m, v_2) and which it casts to the verb with which it occurs most frequently. The votes for each verb are summed over all of the k nearest neighbours of n and the verb with the most votes wins. Performance is measured as error rate.

$$\text{Error rate} = \frac{1}{T}\left(\text{\# of incorrect choices} + \frac{\text{\# of ties}}{2}\right) \tag{5.4}$$

where T is the number of test instances and a tie results when the neighbours cannot decide between the two alternatives.

The usual approach is to optimise the number of nearest neighbours considered, k, as a global parameter. However, we recognise that there may not be an optimal global value of k and that the optimal value may vary from word to word. For example, for some words the ith neighbour may seem very similar whereas for others it may be very different. Unfortunately, due to the shortage of data, it is not possible to optimise k directly for each target noun. However, we can attempt to determine an appropriate value of k by considering how similar the neighbours are to the target noun. In this work, we experiment with two different approaches. In the first, we take the standard approach and optimise k globally. In the second, we optimise a similarity threshold which is used to determine when the next potential neighbour is too dissimilar to be considered.

Other than looking at the effect of how k is determined, our method differs from that of Lee (1999) in how the test set is constructed and in the voting scheme employed. In order to explain the rationale behind our method, we will now describe the approach taken by Lee and discuss a number of statistical biases we observed when using this approach. Finally, we will discuss how our own experimental set-up attempts to minimise these biases.

[3]Ten being less than the minimum number (fourteen) of (possibly) indistinct co-occurrences for any target noun in the original test data.

Lee (1999) randomly selects a portion (20%) of the co-occurrence data as test data and then deletes any instances of co-occurrences from this test data which also occur in the training data (i.e. the other 80%). She then replaces each remaining noun-verb pair (n, v_1) in the test data with a noun-verb-verb triple (n, v_1, v_2) where v_2 is selected such that it is of approximately equal probability to v_1 in the training data. This ensures that an approach which backs off to unigram probabilities will do no better than chance. The test set is then split into five disjoint subsets for cross-validation purposes. For each test instance (n, v_1, v_2), neighbours of n are required to decide which verb was the original co-occurrence, which, by construction, is always v_1. In order to do this, Lee employs a voting scheme where the k nearest neighbours of the noun n each vote for the verb with which it co-occurs the most in the training data. In contrast to our own voting scheme, each neighbour is given a single, equal vote. The number of nearest neighbours considered, k, is a parameter which is optimised and performance is measured as error rate:

We will first consider the voting scheme employed and then the method of test set construction.

5.2.1.1 Voting Scheme

In Lee's approach, each neighbour is given a single vote whereas we give each neighbour a vote weighted by the difference in frequency with which it co-occurs with each verb. In other words, we distinguish between cases where a neighbour occurs with each verb approximately the same number of times and where a neighbour occurs with one verb significantly more times than the other. Further, in a preliminary experiment, we observed that a single vote per neighbour appears to introduce a bias towards v_1. We will now discuss this experiment.

As in Chapter 3, we started with the 2,852,300 lemmatised noun-verb direct-object pairs extracted from the BNC using the RASP parser. We then constructed the training and test sets, for the 1000 most frequent nouns, using the approach of Lee (1999). Neighbour sets for each noun were constructed from the training data using $dist_{\alpha 2}$ and $dist_{L_1}$. The task was then carried out using the single-vote per neighbour voting scheme described above and using a frequency-weighted voting scheme. In the frequency-weighted voting scheme, each neighbour m is given a vote which is equal to the difference in frequencies of the co-occurrences (m, v_1) and (m, v_2). The results are shown in Figure 5.4.

Both voting schemes give a similar pattern of results. Using one-vote-per-neighbour, $dist_{\alpha 2}$ marginally outperforms $dist_{L_1}$ whereas using frequency-weighted voting both measures achieve identical optimum error rates at an identical value of k. However, the error rates using frequency-weighted voting are significantly higher than using one-vote per neighbour, which might at first sight suggest that this is not the better voting scheme. However, before coming to this conclusion, we should consider the error rates at $k = 1000$.

As noted by Lee (1999), all measures obtain the same error rate at $k = 1000$. This is because all neighbours are considered and given a vote which is independent of their similarity, and thus the similarity data is effectively thrown away. Accordingly, the error rate at $k = 1000$ depends solely on the distributions of the verbs v_1 and v_2 and one might expect that it should be (approximately) 0.5 since both verbs have (approximately) equal unigram probability. Considering just a single test instance, we can see that this may not be the case using the one-vote-per-neighbour voting scheme. v_1 can achieve a majority by winning votes with a smaller frequency difference. In this case, a smaller number of neighbours have voted for v_2 but each vote has been won by a larger

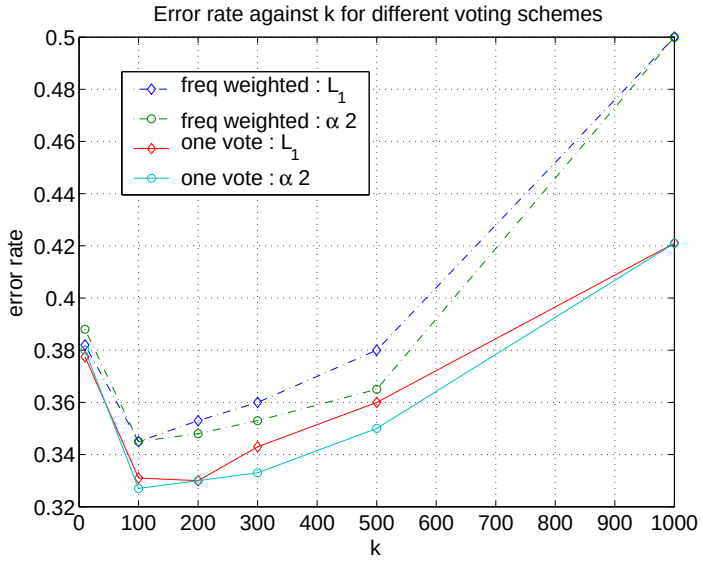

Figure 5.4: Comparison of error rates between one vote per neighbour and frequency weighted voting schemes

frequency difference. Accordingly, in a single test instance v_1 (or conversely v_2) may win at $k = 1000$ using the single-vote-per-neighbour voting scheme. However, over a large number of test instances (23,756 in this experiment), where v_1 and v_2 vary from test instance to test instance, we would expect to achieve an average error rate of approximately 0.5.

However, in our experiments the average error rate obtained at $k = 1000$ using the single-vote-per-neighbour scheme was 0.421 and, similarly, Lee (1999) reports an error rate of 0.383 at $k = 1000$. This is a fairly large deviation from 0.5 and is unlikely to occur by chance. It suggests that there is something in the construction of the test set that biases the evaluation towards the correct choice v_1. When constructing the test set we know that v_1 and v_2 have (approximately) equal frequency in the training set and that v_1 occurs exactly once with n (and that occurrence is in the test set and not in the training set). There is no such restriction on v_2 and therefore we can infer that it is this which causes the bias. Since we know that there is a single (n, v_1) co-occurrence, the distribution of v_1 is likely to be more dispersed (or less selective) than the distribution of the random verb v_2. In other words, v_1 is likely to have more small wins and therefore win more votes overall at $k = 1000$ using the single-vote-per-neighbour voting scheme.

One could attempt to remove this bias by changing the test set construction. In order to do this, the method used would need to place the same restrictions on v_1 and v_2. However, this is difficult since we wish to know that (n, v_1) is more probable than (n, v_2). Our approach, which uses frequency-weighted voting, does not remove the bias but overcomes it by considering the differences in frequencies with which each neighbour co-occurs with each verb.

5.2.1.2 Test Set Construction

We now consider the method of test set construction in more detail.

A main feature of the approach of Lee (1999) is that instances of co-occurrences which occur in the training data are deleted from the test data. Lee's justification for this is that the point of similarity-based estimation is to deal with unseen co-occurrences. However, although we prefer

this approach to one which deletes test instances from the training data (Pereira et al., 1993; Clark & Weir, 2002) since this alternative distorts the training data, we argue that there are remaining problems to be considered.

First, there is no guarantee that the inequality $P(v_1|n) > P(v_2|n)$ holds for the population probabilities. We can ensure that the inequality holds for the sample probabilities by restricting the choice of v_2 such that it is not seen with n in the training data[4]. However, population probabilities cannot be perfectly predicted from sample probabilities; otherwise, probability estimation would be a solved problem. Is a single co-occurrence of (n, v_1) compared to zero co-occurrences of (n, v_2) sufficient evidence to conclude that (n, v_1) is more probable than (n, v_2)?

Second, since the joint probability of n and v_1 must be very small for the co-occurrence pair not to have occurred in the training data, the unigram probability of the verbs to be disambiguated tends to decrease as the unigram probability (or frequency) of the noun increases. When we deleted training co-occurrences from test sets in our experiments, we found significant negative correlation (product-moment correlation coefficient between -0.12 and -0.18) between the noun and verb frequency. Since frequency limits the number of nouns a verb may have occurred with, a low frequency verb will also tend to have occurred with less nouns. This means that the verbs to be disambiguated will have occurred with a lower number of nouns and therefore a smaller number of the target noun's potential neighbours.

The third problem with this test set construction method is that it is biased towards nouns which occur with many different verbs (typically high frequency nouns). In the extreme case, after discarding those pairs which also occur in the training data, there may be no test pairs remaining for a very selective noun. When a test set was constructed using this approach for the 1000 most frequent nouns, we found that 99% of the nouns appeared in the final test set. When we repeated this for a set of 1000 random nouns, the proportion dropped to 60%. Accordingly, using this approach, we are not evaluating over all of the target nouns in a particular set, which could lead us to drawing incorrect conclusions about the nouns in that set.

Our method of test set construction attempts to overcome all three of these problems. We argue that there is no need to discard any test data on the basis of it having been seen in the training data (or vice versa). In the machine-learning paradigm, it is still unseen data which is not used in the training process. Further, by not discarding any seen noun-verb pairs we can be more confident that the correct choice verb, v_1, is more plausible than the incorrect choice verb, v_2. If neighbours can disambiguate correctly between verbs which we are sure occur with the target noun with different probabilities, we can infer that they will also make good choices when the verbs are truly unseen. Further, using this approach, there is no correlation between noun and verb frequency. Finally, we select a proportion of the data for each noun as test data, which means that each noun is treated fairly as to how much data is lost for training purposes whilst allowing us to control the number of test instances for each noun. Accordingly, we are able to remove the bias towards less selective/high frequency nouns.

There is a possible remaining source of bias. This is in how the unigram probabilities of the verbs are estimated. In our preliminary experiments concerned with just the 1000 most frequent nouns, we calculated the unigram probabilities of the verbs over the training data for the target

[4]This is the approach taken by Lee and Pereira (1999) and by Clark and Weir (2002) and in our preliminary experiments in Chapter 3.

nouns, thus ensuring that each had an approximately equal sample probability. However, originally we thought it would be better to calculate the verb probabilities independent of the target nouns being considered. However, this led to further biases towards the correct choice verb v_1. When using the frequency-weighted voting scheme, the error rate at $k = 1000$ was 0.463 and when using a one-vote-per-neighbour voting scheme, the error rate at $k = 1000$ was 0.414. This bizarrely would seem to suggest that a verb which is known to occur once with one of the most frequent nouns is more likely to occur more frequently in total with all of them than a verb of equal unigram probability that is not known to occur with any of the 1000 most frequent verbs. This poses a fairly serious problem for us since it seems fundamentally wrong to us to change our estimates of verb probabilities based on the nouns we are considering as potential neighbours. If we change the set of potential neighbours, for instance if we wish to consider just low frequency or high frequency potential neighbours, then we would need to change the test set, which makes it difficult to make comparisons. If we do not change the test set, however, are our comparisons valid if one verb has a much higher probability of occurring with the set of all potential neighbours than the other verb? This problem serves to highlight the delicacy of pseudo-disambiguation experiments. In the following experiments, we decided to estimate verb probability over all of the direct-object data available so that the test instances for a particular noun would remain constant whatever potential neighbours are considered. However, this means it is necessary to determine the baseline empirically rather than assuming that it is 0.5.

5.2.2 Results

In this section we present results on the pseudo-disambiguation task for all of the CRMs described in Chapter 4. We also compare the results with those for three existing distributional similarity measures and a WordNet-based measure. The distributional similarity measures considered are $dist_{L_1}$ (Section 3.1.1.1), sim_{lin} (Section 3.1.5.2) and $dist_{\alpha 1}$ (Section 3.1.4.3). The WordNet based measure used will be referred to as sim_{wn} and is the one proposed by Lin (1997) and discussed in Section 2.3.6.

As discussed earlier, for the purposes of these experiments, we have two sets of 10,000 unseen test instances. One set contains test instances for the high frequency nouns and the other contains test instances for the low frequency nouns. As discussed in Section 5.2.1, each set is split into five disjoint subsets for five-fold cross-validation purposes. Further, three of the subsets are used as a development set and two as a test set to study the relationship between the parameters and the error rate. In the case of the CRMs, the parameters that are optimised are β, γ and either k (the number of nearest neighbours) or t (the similarity threshold). For the existing measures, the only parameter to be optimised is either k or t, noting that for sim_{lin} and sim_{wn} t is a similarity threshold whereas for $dist_{L_1}$ and $sim_{\alpha 1}$ t is a distance threshold. Of course, it is not feasible to make an exhaustive search for the optimal parameter settings, or even possible when these lie on a continuous scale. We optimised all of the parameters simultaneously and varied the step-size for each parameter in order to obtain a fairly accurate estimate of the optimal value in the development set. The step-sizes used gave us fairly smooth results from which we can interpolate intermediate results with reasonable confidence. The graphs in Figures 5.6 to 5.14 show the points at which performance was measured.

Having determined the optimal parameter settings for each measure, we calculated the performance at these settings using the test set. We now look at the results in detail. We first look at the results for optimising k and then for optimising t.

5.2.2.1 Optimising k, the Number of Nearest Neighbours Considered

The results are summarised in Table 5.2. The table gives the average optimal error rates for each measure calculated using five-fold cross validation. The results are also divided between those for high frequency nouns and low frequency nouns. The cross-validated average optimal error rates are also illustrated in Figure 5.5.

A baseline for these experiments is the performance obtained by a technique that backs-off to the unigram probabilities of the verbs being disambiguated. By construction of the test-set this should be approximately 0.5. The actual empirical figures are 0.553 for the high frequency noun test set and 0.586 for the low frequency noun test set. The deviation from 0.5 is due to the unigram probabilities of the verbs not being exactly equal and to them being calculated over training data for all nouns rather than just the training data for the 2000 target nouns. These baseline error-rates are also different to what is observed when all 1999 potential neighbours are considered. In this case, we obtain an error rate of 0.6885 for the high frequency noun test set and 0.6178 for the low frequency nouns test set. These differences are due to the fact that the correct choice verb, but not the incorrect choice verb, has occurred, possibly many times, with the target noun in the training data but a noun is not considered as a potential neighbour of itself.

Measure	Noun Frequency	
	high	low
	cross-validated error	cross-validated error
sim^{add}_{type}	0.196 (0.005)	0.197 (0.007)
sim^{add}_{tok}	0.219 (0.012)	0.241 (0.007)
sim^{add}_{mi}	**0.178 (0.005)**	**0.169 (0.008)**
sim^{dw}_{type}	0.214 (0.009)	0.185 (0.004)
sim^{dw}_{tok}	0.234 (0.012)	0.202 (0.009)
sim^{dw}_{mi}	0.187 (0.005)	0.176 (0.008)
$dist_{L_1}$	0.234 (0.012)	0.202 (0.009)
sim_{lin}	0.193 (0.006)	0.181 (0.007)
$dist_{\alpha 1}$	0.230 (0.015)	0.192 (0.007)
sim_{wn}	0.295 (0.009)	0.294 (0.004)

Table 5.2: Mean optimal error rates using five-fold cross-validation (when optimising k, γ and β). Figures in brackets are standard deviations in the optimal error rates across the five folds.

From the results in Table 5.3 and Figure 5.5 we can make the following observations:

1. All of the distributional similarity measures perform considerably better than the WordNet-based measure at this task for high and low frequency nouns
2. For the majority of measures, performance is better for low frequency nouns than for high frequency nouns.

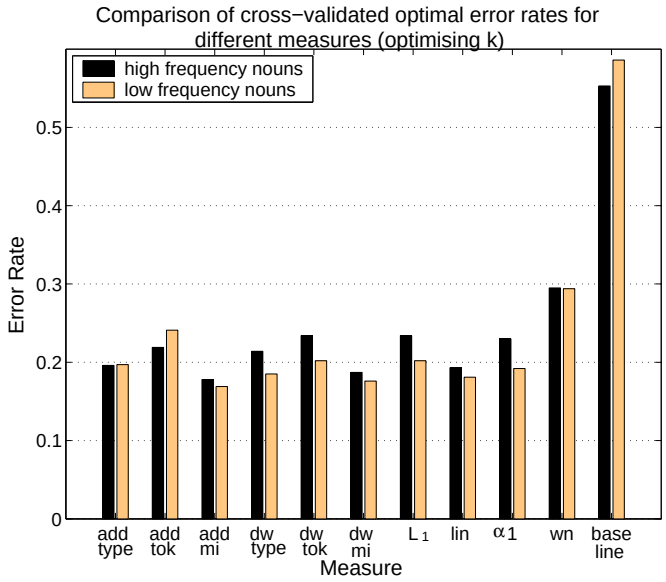

Figure 5.5: Barchart illustrating cross-validated optimal error rates for each measure when k is optimised

3. The two best measures appear to be the additive MI-based model and the difference-weighted MI-based model. These are followed by Lin's MI-based measure.

4. The token-based CRMs perform badly compared to the other CRMs, but comparably to the L_1 Norm (which is of course equivalent to the difference-weighted token-based CRM) and the α-skew divergence measure.

5. The additive models appear to perform best for high frequency nouns whereas the difference-weighted models appear to perform best for low frequency nouns.

We now consider the effects of the different parameters on performance. Since we use the development set to determine the optimal parameters, we consider performance on the development set as each parameter is varied. Table 5.3 shows the optimised parameter settings in the development set, error rate at these settings in the development set and error rate at these settings in the test set. Figure 5.6 shows how performance varies with k for the existing similarity measures. For the CRMs, we consider how the performance varies with each parameter when the other parameters are held constant at their optimum values. Figure 5.7 shows how performance varies with k, Figure 5.8 shows how performance varies with β and Figure 5.9 shows how performance varies with γ. However, we note that there is interaction between the different parameters. In particular, a high value of γ reduces the effect of β and therefore, when the optimal value of γ is high, the variation with respect to β will be less.

We now make the following observations:

1. Variation with respect to k is similar for all models and measures and is what we would expect. To begin with, considering more neighbours increases performance since more neighbours allow decisions to be made in a greater number of cases. However, when k increases beyond an optimal value, a greater number of these decisions will be in the wrong direction since these words are not very similar to the target word, leading to a decrease in performance.

Measure	Noun Frequency									
	high					low				
	optimal parameters			devel. error	test error	optimal parameters			devel. error	test error
	γ	β	k			γ	β	k		
sim^{add}_{type}	0.25	0.8	150	0.1926	0.1933	0.25	0.75	100	0.1916	0.1998
sim^{add}_{tok}	0	0.8	250	0.2107	0.2244	0.5	0.1	130	0.2343	0.2330
sim^{add}_{mi}	**0.25**	**0.8**	**170**	**0.1754**	**0.1856**	**0.5**	**0.8**	**120**	**0.1693**	**0.1781**
sim^{dw}_{type}	0	0.6	50	0.2084	0.2149	0.25	0.3	190	0.1771	0.1882
sim^{dw}_{tok}	n/a	n/a	60	0.2270	0.2335	n/a	n/a	50	0.1944	0.2064
sim^{dw}_{mi}	0.25	0.8	100	0.1809	0.1929	0.5	0.7	160	0.1722	0.1734
$dist_{L_1}$	n/a	n/a	60	0.2270	0.2335	n/a	n/a	50	0.1944	0.2064
sim_{lin}	n/a	n/a	50	0.1895	0.1994	n/a	n/a	80	0.1789	0.1855
$dist_{\alpha 1}$	n/a	n/a	30	0.2279	0.2334	n/a	n/a	60	0.1890	0.1965
sim_{wn}	n/a	n/a	40	0.2937	0.301	n/a	n/a	40	0.2954	0.2921

Table 5.3: Summary of results on pseudo-disambiguation task when optimising k, the number of nearest neighbours

2. Variation with respect to β and γ differs quite a lot between the models. However, in general, a lower value of γ and a higher value of β lead to better results.

3. A low value of γ indicates that a combination of precision and recall which is closer to a weighted arithmetic mean is better than one which is closer to an unweighted harmonic mean. However, the differences observed with respect to γ are fairly small. This suggests that if the optimal weighted combination of precision and recall is not known, then an unweighted harmonic mean is not a bad choice.

4. The best-performing measures are the MI-based models. These perform best at fairly high values of β (around 0.8). This indicates that a potential neighbour with high precision retrieval of informative features is more useful than one with high recall retrieval. In other words, it seems that it is better to sacrifice being able to make decisions on every test instance with a small number of neighbours in favour of not having neighbours that predict incorrect verb co-occurrences.

5. There is very little difference between the results for the additive and difference-weighted MI-based models. This suggests that the difference in extent to which each noun has each feature is fairly irrelevant. It is enough to know that the two nouns can occur with a particular verb. However, the difference-weighted model is slightly more stable with respect to the parameters β and γ. This is because if a potential neighbour has high recall but not high precision retrieval of the target noun's co-occurrences, there will be a greater difference in the extent to which the retrieved features are shared, which will be penalised by the difference-weighted model.

6. The additive type-based CRM also performs better at high values of β. The variation seen with β for this model is a lot greater than for the MI-based models. This suggests that having high recall of uninformative features, i.e. ones shared by many nouns, is not a good indicator of similarity.

7. The token-based CRMs perform significantly worse than the other models. This suggests

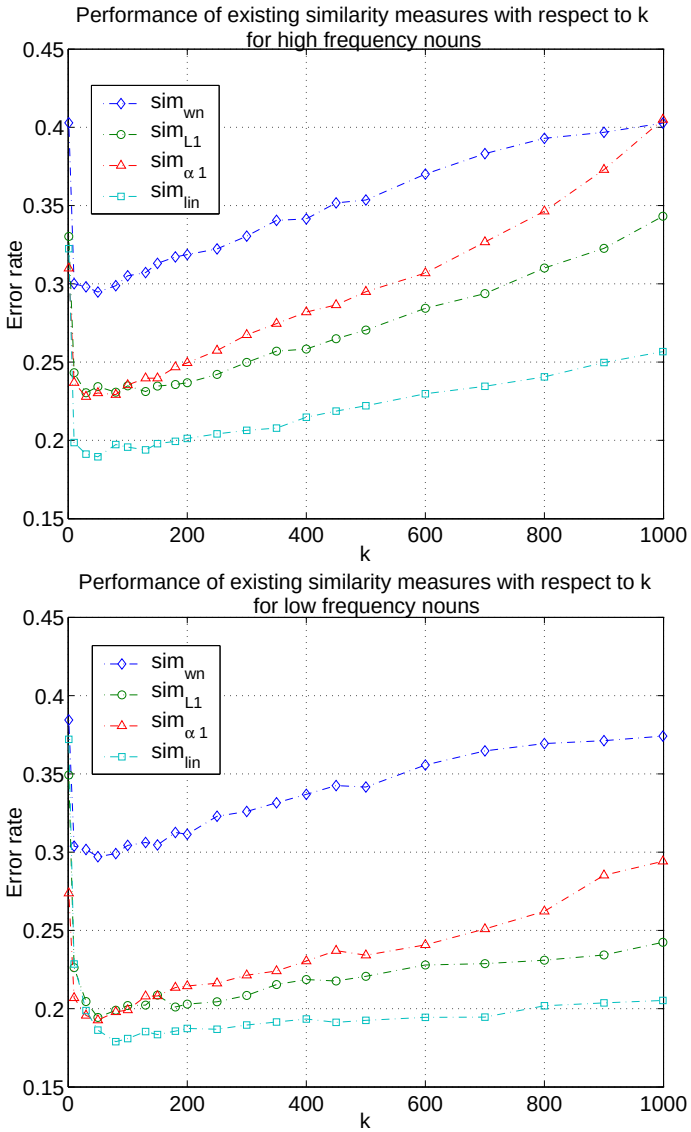

Figure 5.6: Performance of existing similarity measures with respect to *k*

that it is type retrieval, rather than token retrieval, which is important in a distributional similarity measure.

8. The additive token-based model and the difference-weighted type-based model both buck the trend and perform better at lower values of β for low frequency nouns. High recall in the difference-weighted type-based model in fact performs almost as well as high precision in the MI-based models for low frequency nouns. Further, all of the CRMs do better at high values of recall for low frequency target nouns than they do at high levels of recall for high frequency target nouns. Higher recall may do better for low frequency nouns, since high precision neighbours of low frequency nouns will tend to be very infrequently occurring nouns.

5.2.2.2 Optimising t, the Similarity Threshold

In this set of experiments we optimised a similarity or distance threshold rather than the number of nearest neighbours considered. When using a threshold, we consider any neighbour with a greater similarity (or smaller distance) to the target noun than the threshold. When using a similarity

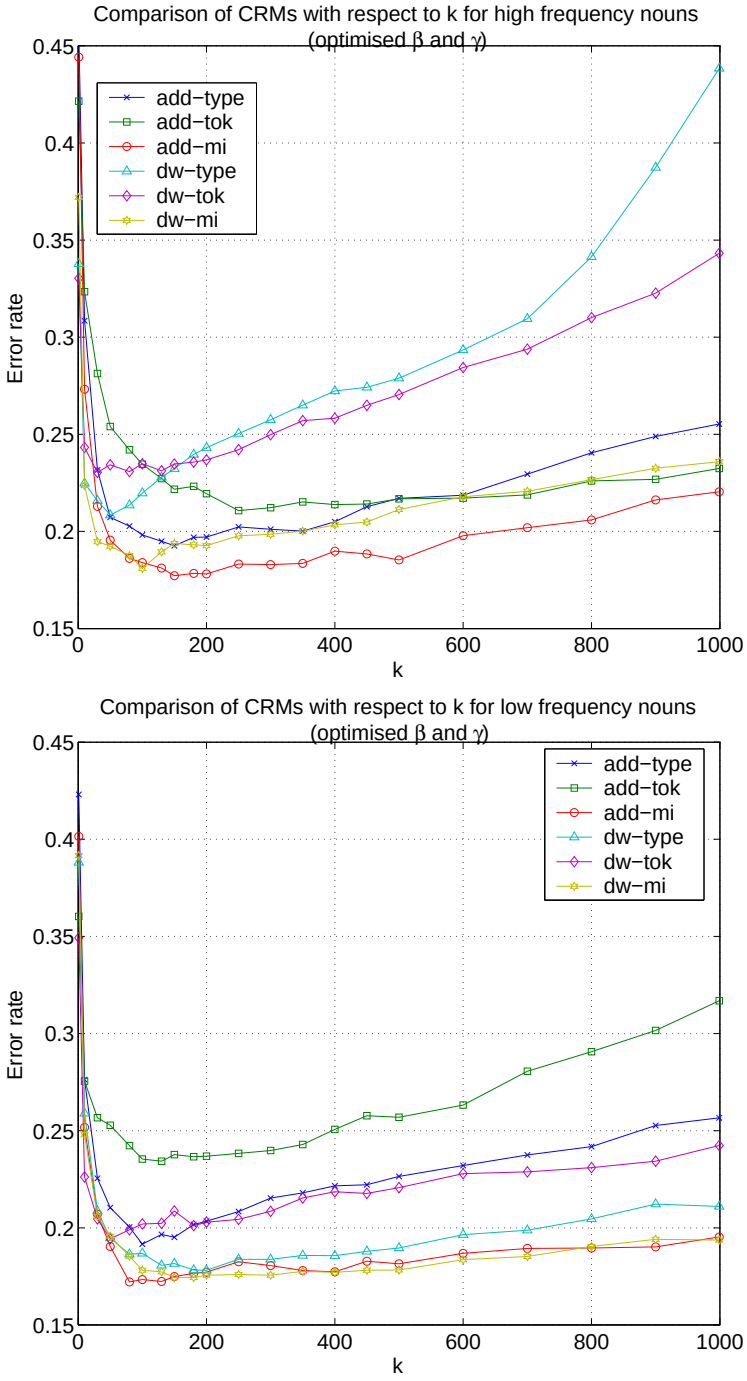

Figure 5.7: Performance of CRMs with respect to *k* (at optimal values of β and γ)

threshold, i.e., with true similarity measures, increasing the threshold leads to less neighbours being considered. When using a distance threshold, i.e., with distance measures, increasing the threshold results in more neighbours being considered.

The results are summarised in Table 5.5. As before, the table gives the cross-validated mean optimal error rates and corresponding standard deviations for each measure. These results are also divided between those for high frequency nouns and low frequency nouns. The cross-validated average optimal error rates are illustrated in Figures 5.10.

From these results, the following observations can be made:

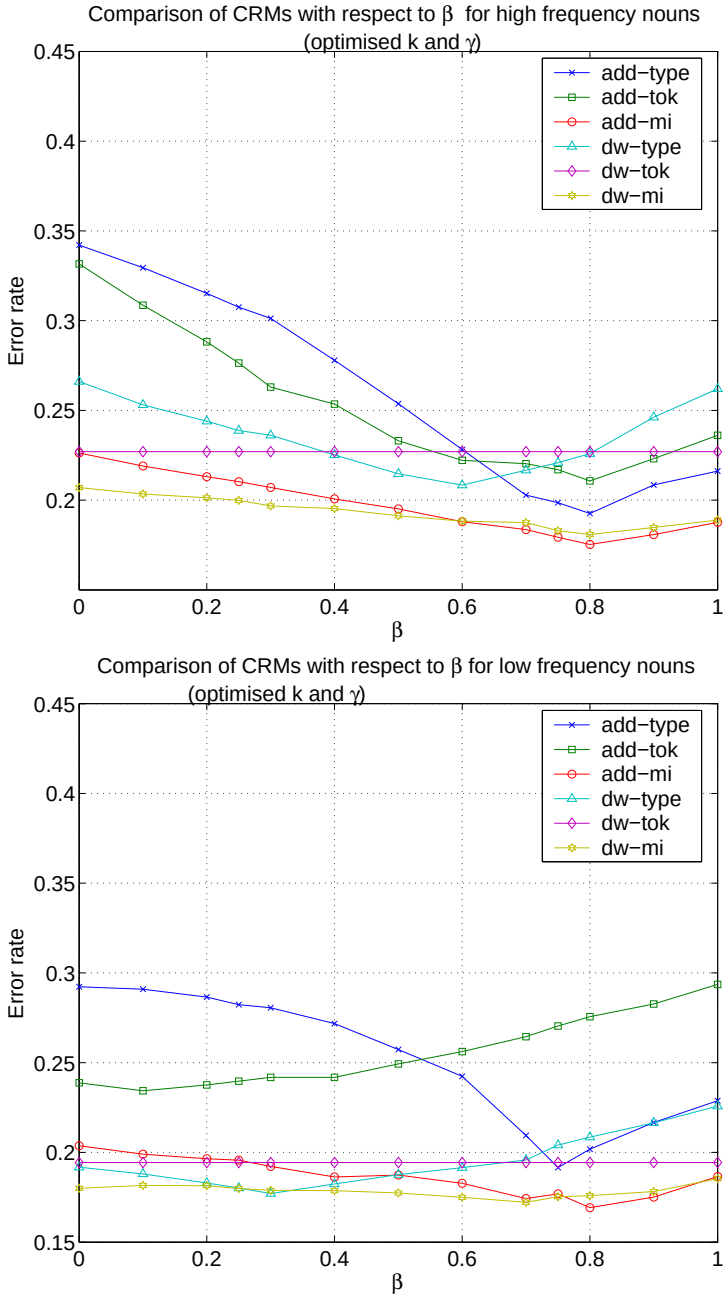

Figure 5.8: Performance of CRMs with respect to β (at optimal values of k and γ)

1. In relative terms, the pattern of results is the same as when k is optimised.
2. The absolute values of the optimal error rates obtained are higher (i.e. worse) than when k is optimised.

As before, we now look at the effect of the different parameters on performance. Table 5.5 gives the optimal parameter settings in the development set, error rate in the development set at these settings and the error rate in the test set at these settings. Figure 5.11 shows how performance varies with t for the existing similarity measures. In this case, we split the graphs between the similarity measures (sim_{lin} and sim_{wn}) and the distance measures ($dist_{L_1}$ and $dist_{\alpha 1}$). For the CRMs, we consider how performance varies with each parameter when the other parameters are

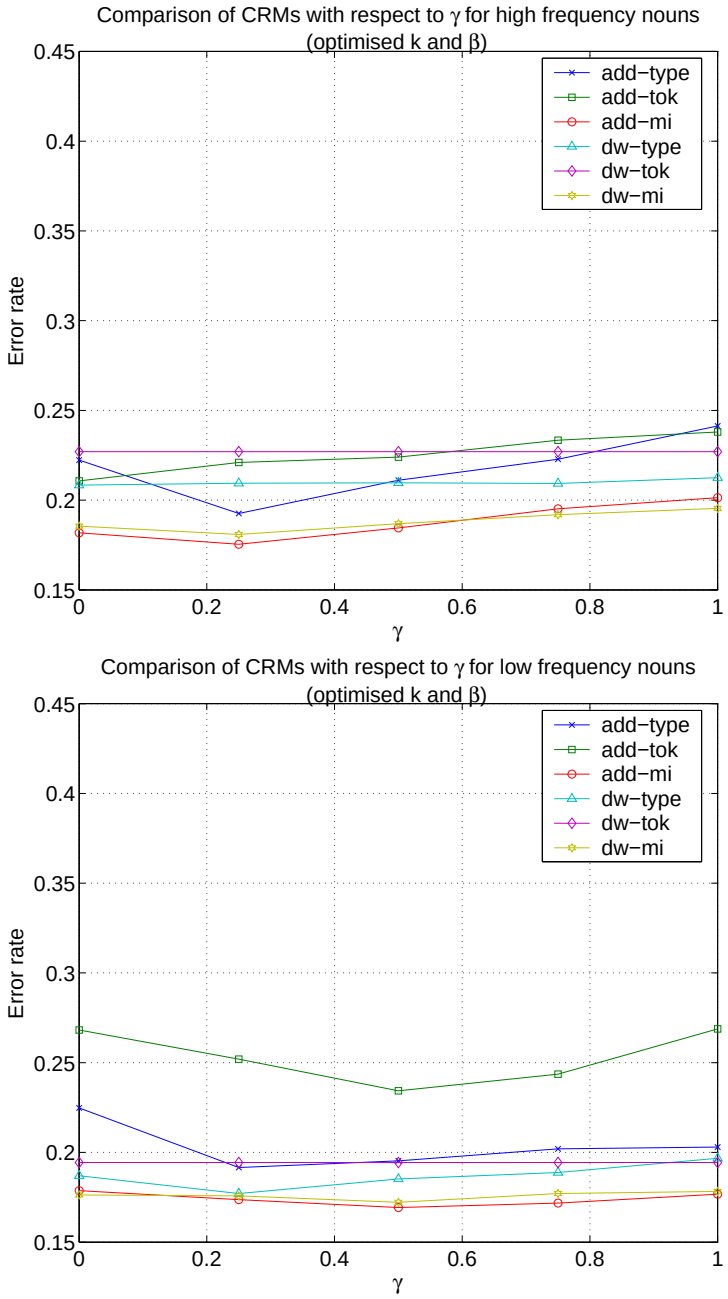

Figure 5.9: Performance of CRMs with respect to γ (at optimal values of *k* and β)

held constant at their optimum values. Figure 5.12 shows how performance varies with *t*, Figure 5.13 shows how performance varies with β and Figure 5.14 shows how performance varies with γ.

From these graphs, we can make the following observations:

1. Regarding the CRMs, the patterns with respect to β and γ are similar to those observed when *k* was optimised.

2. The most striking feature of the graphs is the differences amongst measures in the optimal threshold values. This value reflects the average similarity of a target noun's nearest neighbours. For example, using the MI-based measures very few pairs of words have a similarity of greater than around 0.25. This is reflected in the sharp gradient change observed around this point. The WordNet based measure, on the other hand, tends to find the most similar

Measure	Noun Frequency	
	high	low
	cross-validated error	cross-validated error
sim^{add}_{type}	0.218 (0.006)	0.202 (0.008)
sim^{add}_{tok}	0.237 (0.012)	0.236 (0.004)
sim^{add}_{mi}	**0.187 (0.008)**	**0.171 (0.004)**
sim^{dw}_{type}	0.214 (0.004)	0.186 (0.009)
sim^{dw}_{tok}	0.230 (0.006)	0.208 (0.010)
sim^{dw}_{mi}	0.199 (0.007)	0.176 (0.006)
$dist_{L_1}$	0.214 (0.004)	0.208 (0.010)
sim_{lin}	0.199 (0.006)	0.189 (0.008)
$dist_{\alpha 1}$	0.232 (0.008)	0.201 (0.009)
sim_{wn}	0.299 (0.006)	0.309 (0.007)

Table 5.4: Mean optimal error rates using five-fold cross-validation (when optimising t, γ and β). Figures in brackets are standard deviations.

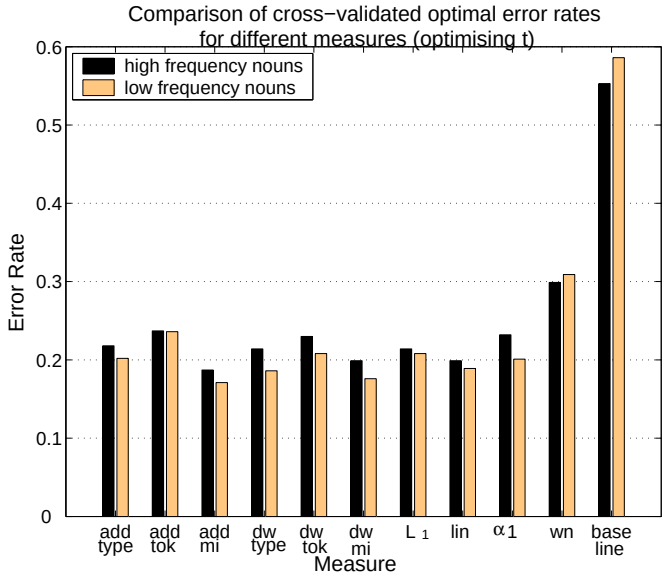

Figure 5.10: Barchart illustrating cross-validated optimal error rates for each measure when t is optimised

pairs of words very similar (above 0.9). Accordingly, the optimal similarity threshold for this measure is a lot higher than for the MI-based measures.

Measure	Noun Frequency									
	high					low				
	optimal parameters			devel. error	test error	optimal parameters			devel. error	test error
	γ	β	k			γ	β	k		
sim^{add}_{comb}	0.5	0.6	0.45	0.2152	0.2147	0.25	0.8	0.18	0.1974	0.2031
sim^{add}_{prob}	0	0.7	0.65	0.2291	0.2446	0.25	0.5	0.4	0.2336	0.2353
sim^{add}_{mi}	**0**	**0.75**	**0.17**	**0.1833**	**0.1891**	**0**	**0.9**	**0.07**	**0.1708**	**0.1724**
sim^{dw}_{comb}	0.5	0.6	0.35	0.2104	0.2180	0.25	0.4	0.14	0.1843	0.1850
sim^{dw}_{prob}	n/a	n/a	0.35	0.2307	0.2281	n/a	n/a	0.2	0.2056	0.2087
sim^{dw}_{mi}	0.75	0.6	0.15	0.1889	0.2006	0.25	0.75	0.06	0.1738	0.172
$dist_{L_1}$	n/a	n/a	1.3	0.2307	0.2281	n/a	n/a	1.6	0.2056	0.2087
sim_{lin}	n/a	n/a	0.16	0.1953	0.2039	n/a	n/a	0.11	0.181	0.1901
$dist_{\alpha 1}$	n/a	n/a	2.2	0.2310	0.2384	n/a	n/a	2.5	0.2005	0.2026
sim_{wn}	n/a	n/a	0.75	0.2974	0.299	n/a	n/a	0.7	0.2991	0.3088

Table 5.5: Summary of results of pseudo-disambiguation task when optimising t, the similarity threshold

5.2.3 Correlation between Pseudo-Disambiguation Results and Smoothing Results

In order to investigate whether higher performance on the pseudo-disambiguation task really predicts higher performance in smoothing, we repeated the smoothing experiments in Section 5.1 using the best-performing measure from the pseudo-disambiguation experiments. We will now discuss the modifications to the experimental set-up and then discuss our results.

5.2.3.1 Experimental Set-up

We took the training data described in Section 5.1 and calculated the similarities between all pairs of nouns using sim^{add}_{mi} and parameter settings ($\gamma = 0.25$, $\beta = 0.8$). We then smoothed the distributions using the smoothing algorithm described in Section 5.1.2. Since, sim^{add}_{mi} directly defines a similarity function, a different neighbour weighting function was required. For these experiments, we tried:

$$\text{weight}_1(N_i) = sim^{add}_{mi}(N_i, C_0) \tag{5.5}$$

and

$$\text{weight}_2(N_i) = (sim^{add}_{mi}(N_i, C_0))^2 \tag{5.6}$$

With regard to the stopping condition, we just implemented stopping conditions 1 and 3, since these gave the best results in Section 5.1. However, we continued to use $dist_{\alpha 1}$ to measure the distance between the original distribution and the derived distribution and also to measure the distance between the derived distribution and the unseen empirical distribution, so that we could make a fairer comparison of the effect of deriving nearest neighbours using a different similarity measure.

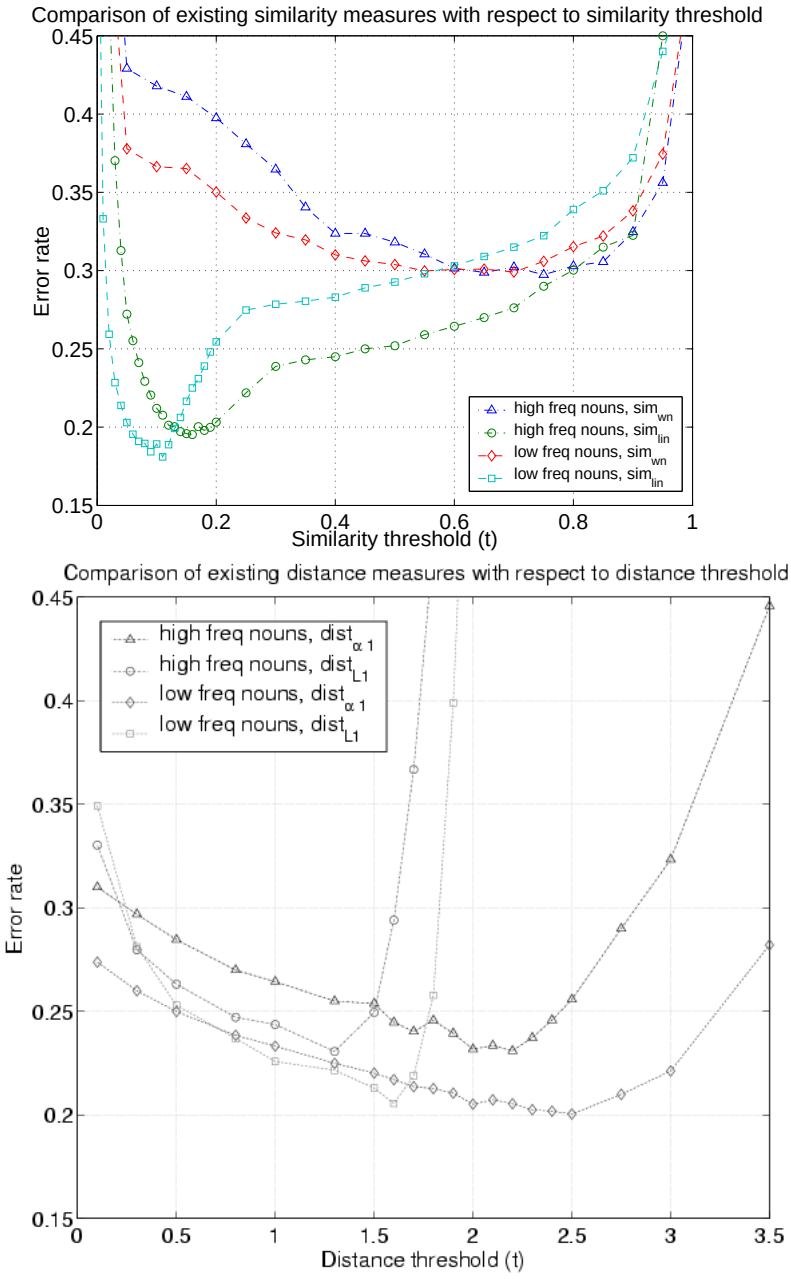

Figure 5.11: Performance of existing similarity measures with respect to t

5.2.3.2 Results

In this experiment, we were able to obtain optimal results using stopping condition 3 ($\kappa = 1.5$) and the $weight_2$ function. With these settings, mean distance from the test set distributions dropped from 0.815 ($\sigma = 0.270$) to 0.786 ($\sigma = 0.276$) whilst mean distance from the training set distributions dropped from 0.238 ($\sigma = 0.058$) to 0.209 ($\sigma = 0.044$). The increase in performance over the using the α-skew divergence measure is statistically significantly (at all levels). In terms of coverage, we achieved 90.5% type coverage and 97.6% token coverage. This is again considerably higher than when using the α-skew divergence measure which achieved 83.6% type coverage and 95.7% token coverage.

However, for this measure, the choice of the neighbour weighting function seemed more sig-

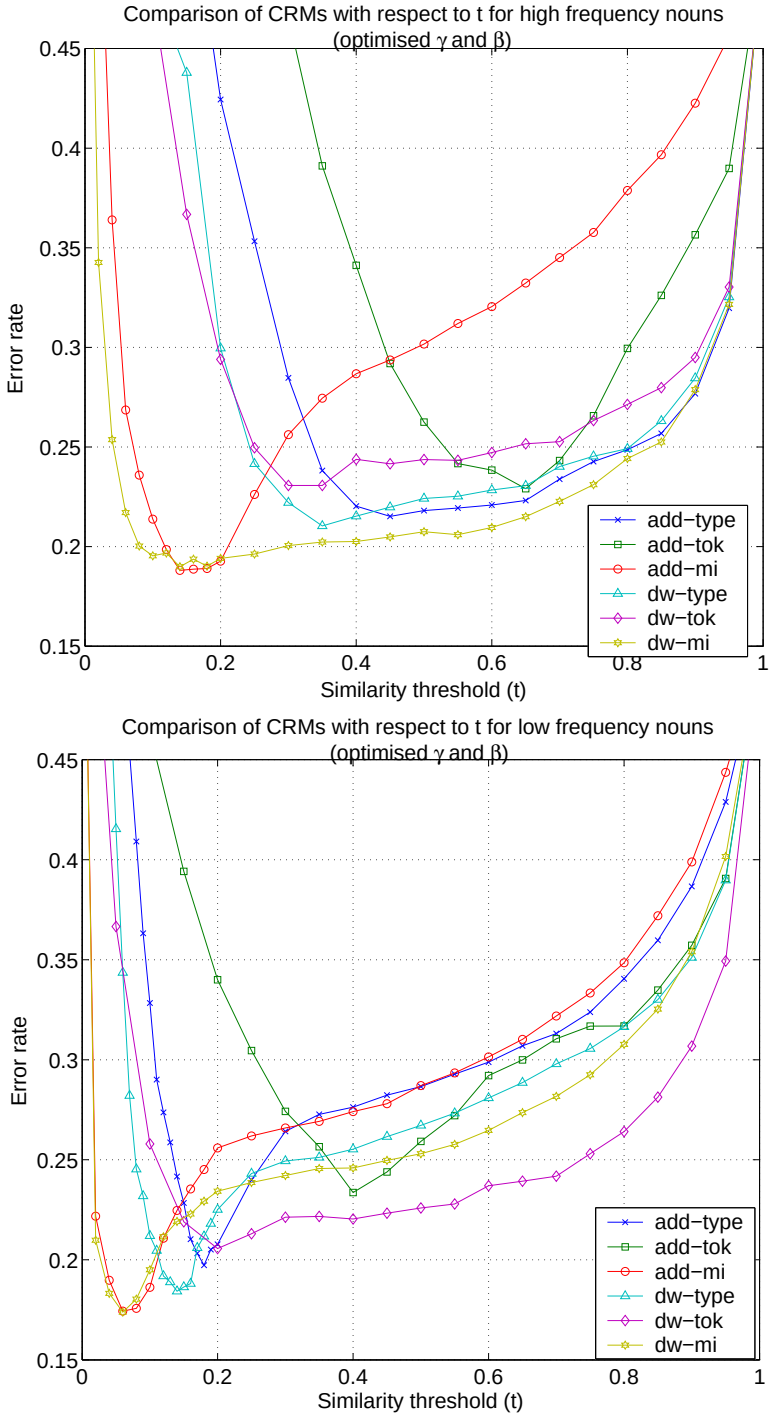

Figure 5.12: Performance of CRMs with respect to t (at optimal values of β and γ)

nificant than when using the α-skew divergence measure. Results using the $weight_1$ function led to performance levels that were not significantly better than when using $dist_{\alpha 1}$.

From this we conclude that the use of the pseudo-disambiguation task does appear to be a valid way of predicting performance on a more complex language modelling task. However, to actually implement smoothing, more careful consideration of the smoothing algorithm and the weight function used would be required. This is an area for further research.

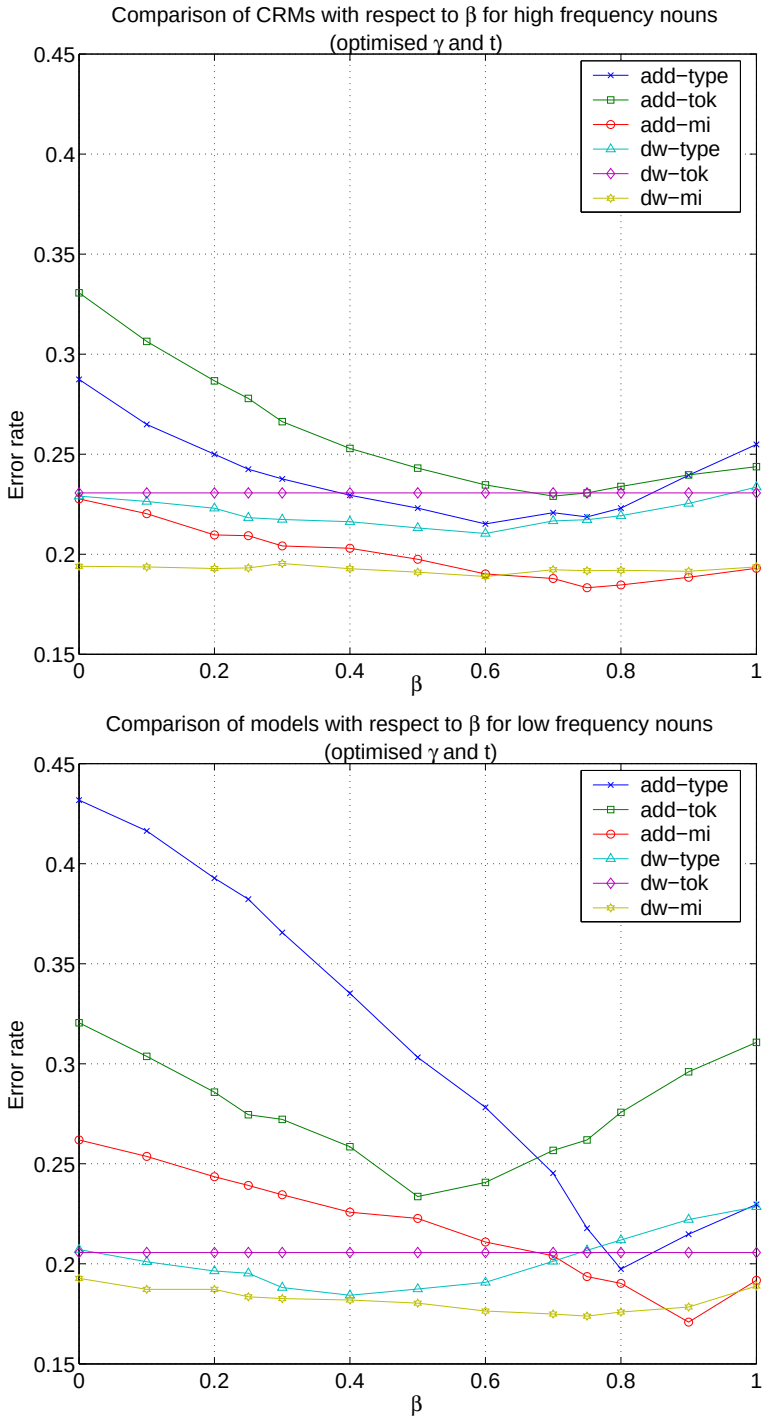

Figure 5.13: Performance of CRMs with respect to β (at optimal values of t and γ)

5.2.4 Discussion

In this section we have seen that neighbours that are similar in terms of co-occurrences are better at predicting other co-occurrences than neighbours that have high semantic similarity with the target noun. This remains true even for low frequency target nouns for which there is fairly little co-occurrence data. In fact, the distributional similarity measures tend to perform even better for these low frequency nouns than for high frequency nouns. One possible explanation for this is that a low frequency word may have only one attested sense in the data. In contrast, high frequency

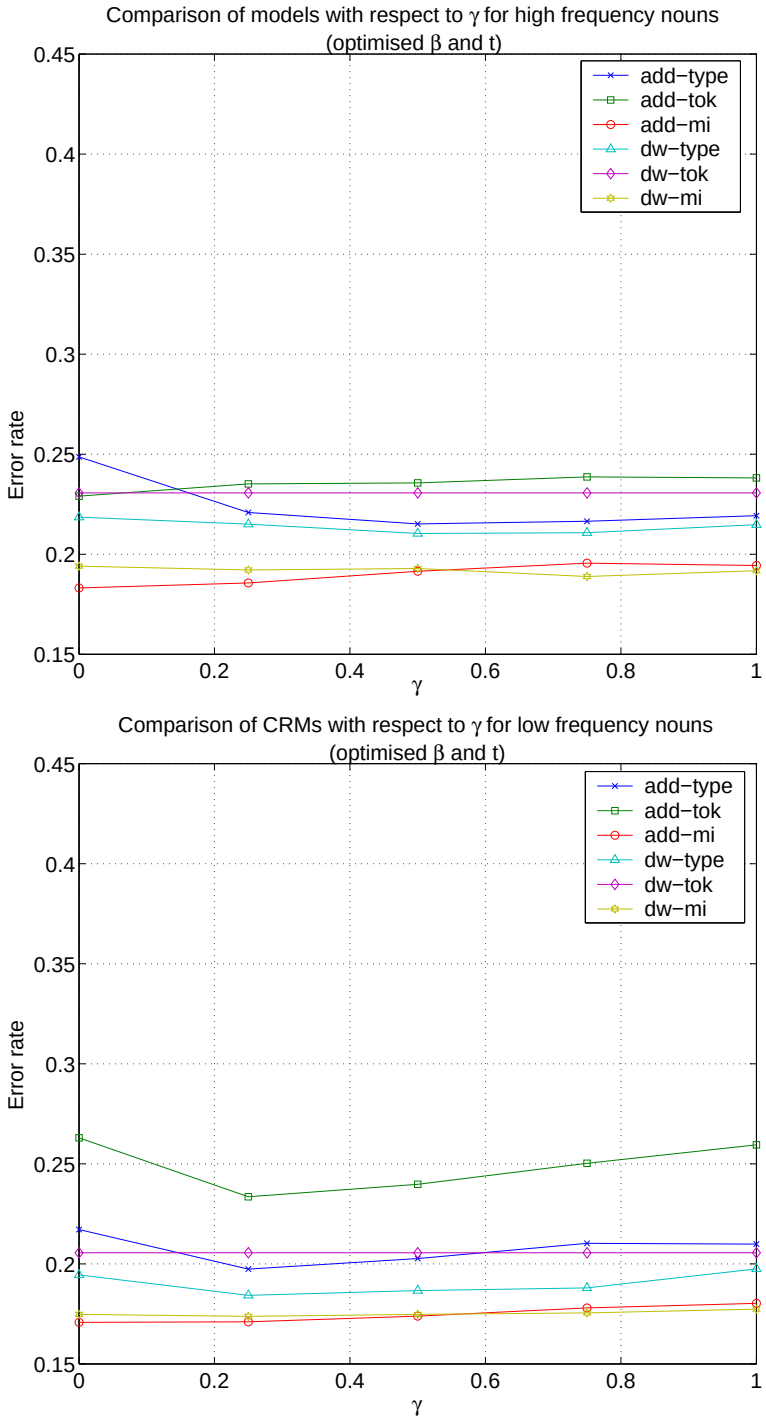

Figure 5.14: Performance of CRMs with respect to γ (at optimal values of t and β)

words are likely to have more attested senses. This highlights one of the benefits of using a thesaurus derived from a sample of the language to be modelled.

We have also shown that using a model in which nearest neighbours have high precision retrieval of the most informative features is better than using the other models considered here. Further, this high performance is obtained regardless in the difference in extent to which each word has each feature.

We expected to find that optimising a similarity threshold would lead to improved results over

optimising the number of nearest neighbours considered. However, this is not borne out by the empirical evidence. It would seem that words vary as much, if not more, in the distance at which another word should be considered a neighbour as in the number of "close" neighbours they have.

5.3 Conclusions

In this chapter we have considered the application of distributional similarity measures in language modelling. We have shown, as in previous work by Dagan et al. (1999), that it is plausible to smooth co-occurrence probability distributions using nearest neighbour techniques. The smoothed distributions obtained are substantially closer to unseen empirical distributions than are the MLE distributions or distributions derived using Add1 type smoothing. Further, by not considering more neighbours once a certain condition is met, we achieve improved results and efficiency gains. The most effective stopping condition considered the amount of change to the original distribution as a function of word frequency.

We then considered the use of pseudo-disambiguation experiments to evaluate different distributional similarity measures. We saw that the experimental set-up in a pseudo-disambiguation task is very delicate and it seems virtually impossible to eliminate all possible biases from the task. However, we developed an approach intended to minimise these biases.

Using the pseudo-disambiguation task we have been able to establish properties of distributional similarity measures which might be desirable in nearest neighbour smoothing techniques. We can now attempt to answer the three questions outlined in Chapter 2 (in the context of this application). First, it appears that contexts in which a noun occurs more frequently than would be expected if nouns and contexts occurred independently are the most salient contexts for determining distributional similarity. In other words, more selective verbs do tell us more about the similarity of their arguments than less selective verbs, even if the nouns occur with the less selective verbs more frequently. Second, the difference in extent to which each word occurs in each context does not appear to be a major factor. It appears to be sufficient to know that each word has occurred in that context. Finally, a measure that can exploit the asymmetry in the concept of substitutability performs better than one that requires similarity to be symmetric. In this task, using the best performing models, neighbours that have high precision retrieval of the target noun's co-occurrences performed better than neighbours that have high recall retrieval of the target noun's co-occurrences.

However, language modelling is not the only potential application for distributional similarity techniques. In the next chapter we consider what properties of distributional similarity measures might be desirable in the semantic domain.

Chapter 6

Predicting Semantic Similarity

In this chapter, we consider the ability of distributional similarity to predict semantic similarity. We do this by comparing corpus-derived distributional similarity judgements with human semantic similarity judgements and by carrying out a task which requires semantic similarity information. We have two sources of human semantic similarity data. The first is the human synonymy judgements of Miller and Charles (1991), which now form a standard test set for semantic measures, and which will be henceforth referred to as the M&C data. The second is the manually constructed noun hierarchy of WordNet (Fellbaum, 1998). Finally, the application-based task we consider is that of spelling correction as described by Budanitsky and Hirst (2001).

6.1 Correlation with Human Synonymy Judgements

In this section we evaluate distributional similarity measures against the M&C data, that is, the human synonymy judgements obtained by Miller and Charles (1991).

6.1.1 Methodology

Rubenstein and Goodenough (1965) established human synonymy judgements for 65 pairs of nouns. All of the nouns were from everyday English and the pairs ranged from highly synonymous (e.g. *car-automobile*) to semantically unrelated (e.g. *noon-string*). 51 subjects were asked to rate the pairs according to their "similarity in meaning" on a scale of 4.0 (absolute synonymy) to 0.0 (semantically unrelated). Subsequently, Miller and Charles (1991) repeated the experiment, consulting 38 human judges, for 30 of the original pairs. The 30 were selected such that 10 had been considered highly synonymous (scoring between 3 and 4), 10 had been considered to be of intermediate synonymy (scoring between 1 and 3) and 10 had been considered to be of low similarity (scoring between 0 and 1) in the original experiment. The human synonymy judgements from this experiment (shown in 6.2) have become a standard data set for evaluating semantic similarity measures (Resnik, 1995; Jiang & Conrath, 1997; Lin, 1998c; Budanitsky & Hirst, 2001; Jarmasz & Szpakowicz, 2003).

A similarity measure can be evaluated against the human synonymy judgements by correlating the two sets of similarity scores. Values of Pearson's product moment correlation coefficient ob-

tained by Budanitsky and Hirst (2001) (B&H) and Jarmasz and Szpakowicz (2003) (J&S) for five WordNet based semantic similarity measures are shown in Table 6.1. Other researchers (Resnik, 1995; Jiang & Conrath, 1997; Lin, 1998c) have reported similar values. As discussed by Budanitsky and Hirst (2001), the differences that occur are due to minor differences in implementation, different versions of WordNet and differences in corpora used to obtain probability information. For example, WordNet version 1.6 is used in our own research and in that of Budanitsky and Hirst (2001), whereas Jarmasz and Szpakowicz (2003) use WordNet version 1.7.

Similarity Measure	B&H	J&S
Resnik (1995)	0.774	0.787
Lin (1997)	0.829	0.846
Jiang and Conrath (1997)	0.850	0.696
Hirst and St-Onge (1998)	0.744	0.689
Leacock and Chodorow (1998)	0.816	0.832

Table 6.1: Correlations between WordNet based similarity measures and M&C data according to Budanitsky and Hirst (2001) and Jarmasz and Szpakowicz (2003)

We have performed the same evaluation for a number of distributional similarity measures. These are the CRMs proposed in Chapter 4, Lin's distributional similarity measure (sim_{lin}), the α-skew divergence measure ($dist_{\alpha}$) and the L_1 Norm ($dist_{L_1}$). We also include results for Lin's WordNet based measure (sim_{wn}) to show consistency of our implementation with the work of earlier researchers.

For each measure, we computed the similarity (or distance) between each pair of nouns listed in Table 6.2. We then calculate Pearson's product-moment correlation coefficient (see Section 3.1.2.1) between the set of scores according to the similarity measure and the set of human synonymy judgements. Under the hypothesis that distributional similarity can be used to predict semantic similarity, we would expect to find high positive correlation between the distributional similarity scores and the human synonymy judgements. For those measures that are actually distance measures, we would expect to find high negative correlation between the distributional similarity scores and the human synonymy judgements.

6.1.2 Results

Table 6.2 shows the similarity and correlation values for the existing similarity measures. Table 6.3 shows the similarity values and correlation values for the optimal parameter settings for each CRM. The pairs of words in each table are ordered by their degree of synonymy in the M&C data. The first column shows the human synonymy judgement between the pair of words in the M&C data. Further columns show the distributional similarity (or distance) score for that pair of words. The final row shows the degree of correlation between the similarity scores in that column and in the first M&C column.

Figures 6.1, 6.2, 6.3, 6.4 and 6.5 illustrate the correlation values for each of the models at different parameter settings[1]. To illustrate the significance of the correlation values, Figure 6.6 is

[1]The difference-weighted token-based model is not shown since the results are independent of the parameter values for this model.

a scatter graph for the WordNet-based measure similarity values, Figure 6.7 is a scatter graph for the additive MI-based model's similarity values at $\gamma = 0.75$, $\beta = 1.0$ and Figure 6.8 is a scatter graph for the α-skew divergence measure ($dist_{\alpha 1}(A,B)$) distance values.

Noun pair (n_1-n_2)	M&C	sim_{wn}	sim_{lin}	$dist_{\alpha 1}(n_2,n_1)$	$dist_{\alpha 1}(n_1,n_2)$	sim_{L1}
car-automobile	3.92	1.0	0.04	4.18	3.05	1.87
gem-jewel	3.84	1.0	0.13	2.79	2.98	1.48
journey-voyage	3.84	0.83	0.24	1.82	1.37	1.09
boy-lad	3.76	0.94	0.18	1.97	1.40	1.02
coast-shore	3.70	0.95	0.20	2.57	2.09	1.34
asylum-madhouse	3.61	0.98	0.06	4.43	3.97	1.95
magician-wizard	3.50	1.00	0.10	3.00	2.82	1.52
midday-noon	3.42	1.0	0.08	4.02	4.00	1.75
furnace-stove	3.11	0.19	0.11	3.23	3.46	1.73
food-fruit	3.08	0.29	0.21	1.56	1.61	1.10
bird-cock	3.05	0.63	0.06	2.92	2.32	1.37
bird-crane	2.97	0.59	0.04	3.31	2.54	1.55
tool-implement	2.95	0.92	0.05	3.68	2.98	1.65
brother-monk	2.82	0.27	0.12	3.05	2.36	1.47
crane-implement	1.68	0.35	0.04	3.35	3.62	1.64
lad-brother	1.66	0.28	0.15	1.68	2.12	1.22
journey-car	1.16	0.0	0.05	2.71	3.06	1.6
monk-oracle	1.10	0.26	0.06	3.52	3.70	1.75
cemetery-woodland	0.95	0.16	0.04	4.01	4.04	1.81
food-rooster	0.89	0.11	0.01	4.06	3.04	1.83
coast-hill	0.87	0.63	0.16	2.68	2.92	1.56
forest-graveyard	0.84	0.16	0.08	3.83	3.28	1.71
shore-woodland	0.63	0.16	0.13	3.79	3.51	1.76
monk-slave	0.55	0.28	0.10	3.07	3.27	1.55
coast-forest	0.42	0.15	0.12	3.13	3.35	1.60
lad-wizard	0.42	0.28	0.04	3.46	2.92	1.57
chord-smile	0.13	0.31	0.04	3.76	3.59	1.77
glass-magician	0.11	0.09	0.01	3.95	3.12	1.78
rooster-voyage	0.08	0.0	0.06	2.53	3.25	1.46
noon-string	0.08	0.0	0.0	3.97	4.45	1.96
Correlation with M&C	1.00	0.83	0.45	-0.26	-0.48	-0.39

Table 6.2: Comparison of existing similarity measures against the M&C Data

We first note that since we are evaluating synonymy judgements for pairs of nouns, there is no obvious direction in a particular pairing. However, the asymmetric measures give different results depending on which way the pair is considered. For the α-skew divergence measure, we have considered the results when the first noun in every pair is considered to be the target noun (i.e.

Noun pair	M&C	sim^{add}_{type}	sim^{add}_{tok}	sim^{add}_{mi}	sim^{dw}_{type}	sim^{dw}_{tok}	sim^{dw}_{mi}
Best parameters							
γ	n/a	0.8	0.5	0.75	0.0	-	0.0
β	n/a	1.0	1.0	1.0	1.0	-	1.0
car-automobile	3.92	0.21	0.53	0.19	0.08	0.06	0.23
gem-jewel	3.84	0.14	0.29	0.08	0.18	0.26	0.09
journey-voyage	3.84	0.23	0.59	0.21	0.28	0.46	0.30
boy-lad	3.76	0.24	0.58	0.16	0.32	0.49	0.22
coast-shore	3.70	0.18	0.45	0.15	0.27	0.33	0.24
asylum-madhouse	3.61	0.12	0.15	0.055	0.03	0.02	0.07
magician-wizard	3.50	0.15	0.32	0.058	0.17	0.24	0.07
midday-noon	3.42	0.05	0.10	0.04	0.07	0.13	0.06
furnace-stove	3.11	0.13	0.23	0.06	0.14	0.14	0.07
food-fruit	3.08	0.29	0.62	0.15	0.38	0.45	0.20
bird-cock	3.05	0.16	0.42	0.08	0.21	0.313	0.14
bird-crane	2.97	0.16	0.45	0.08	0.17	0.23	0.10
tool-implement	2.95	0.14	0.33	0.06	0.12	0.17	0.06
brother-monk	2.82	0.18	0.42	0.11	0.23	0.27	0.16
crane-implement	1.68	0.10	0.16	0.020	0.14	0.18	0.03
lad-brother	1.66	0.15	0.44	0.07	0.17	0.39	0.08
journey-car	1.16	0.13	0.38	0.03	0.14	0.20	0.02
monk-oracle	1.10	0.09	0.19	0.05	0.15	0.12	0.06
cemetery-woodland	0.95	0.06	0.10	0.03	0.08	0.09	0.04
food-rooster	0.89	0.17	0.44	0.04	0.08	0.08	0.04
coast-hill	0.87	0.14	0.34	0.09	0.17	0.22	0.10
forest-graveyard	0.84	0.14	0.27	0.09	0.16	0.15	0.14
shore-woodland	0.63	0.14	0.21	0.09	0.14	0.12	0.12
monk-slave	0.55	0.13	0.23	0.06	0.18	0.23	0.09
coast-forest	0.42	0.14	0.22	0.07	0.18	0.20	0.09
lad-wizard	0.42	0.08	0.29	0.04	0.12	0.22	0.07
chord-smile	0.13	0.08	0.21	0.02	0.08	0.12	0.02
glass-magician	0.11	0.17	0.41	0.03	0.11	0.11	0.01
rooster-voyage	0.08	0.07	0.22	0.03	0.08	0.27	0.03
noon-string	0.08	0.02	0.02	0.0	0.02	0.02	0.0
Correlation with M&C	1.00	0.52	0.46	**0.62**	0.42	0.39	**0.61**

Table 6.3: Comparison of CRMs against the M&C Data

$dist_{\alpha 1}(n_2, n_1)$ or $dist_{\alpha 2}(n_1, n_2)$) and also when the second noun in every pair is considered to be the target noun (i.e. $dist_{\alpha 1}(n_1, n_2)$ or $dist_{\alpha 2}(n_2, n_1)$). Of course, it would be possible to consider multiple different combinations. For example, when the first noun is the target noun in the first pair and the second noun is the target noun in the second pair and so on. However, this would

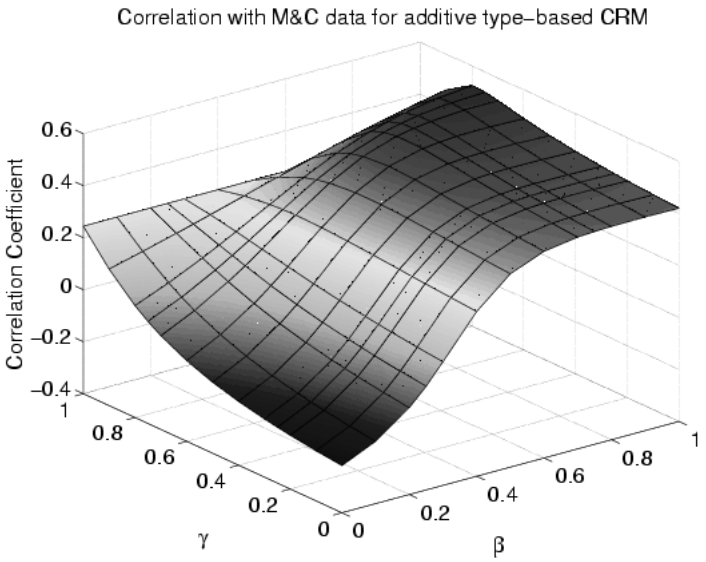

Figure 6.1: Correlation with M&C data for the additive type-based CRM

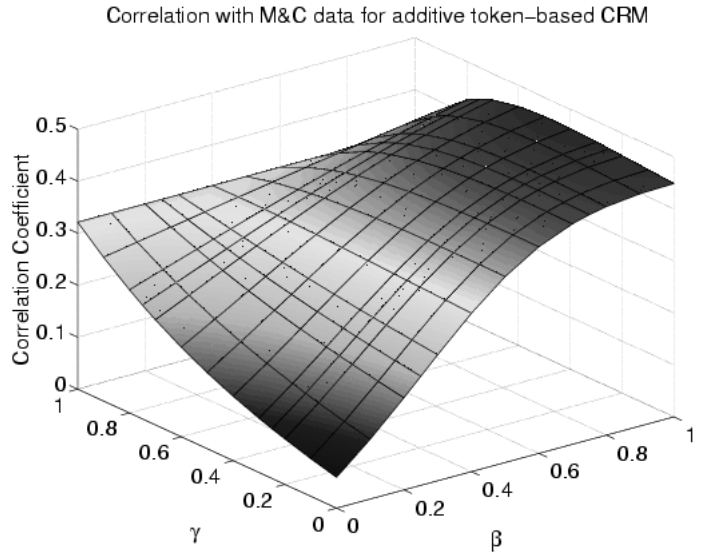

Figure 6.2: Correlation with M&C data for additive token-based CRM

allow an optimisation which would be unfair on the other measures given the small number of pairs being considered. Accordingly, we have considered just the two different cases. We will discuss later why one direction does so much better than the other.

With regard to the CRMs, we have presented results in one direction only. This is where the first noun is considered to be the target noun. We can do this because, although the measure is asymmetric, it is symmetric with respect to the parameter β. In other words, the similarity scores (and thus the correlation) obtained with parameter setting (γ,β) when the first noun is considered to be the target noun are identical to those obtained at (γ,$1-\beta$) when the second noun is considered to be the target noun.

From the results we make the following observations:

1. There is positive correlation between the distributional similarity measures and the human

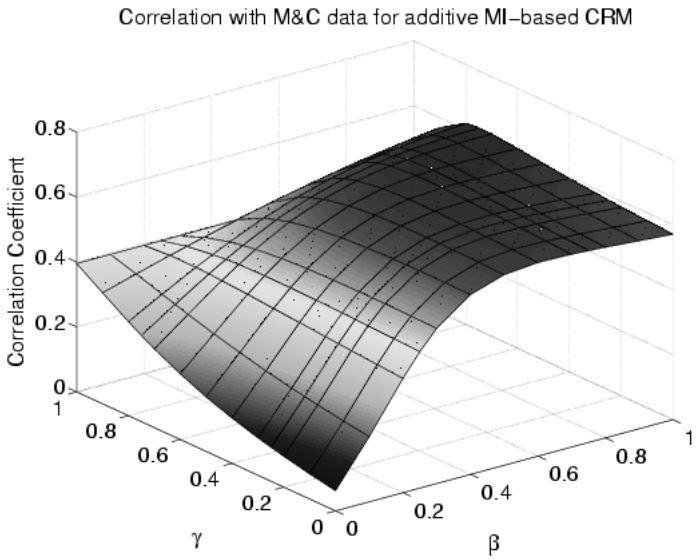

Figure 6.3: Correlation with M&C data for additive MI-based CRM

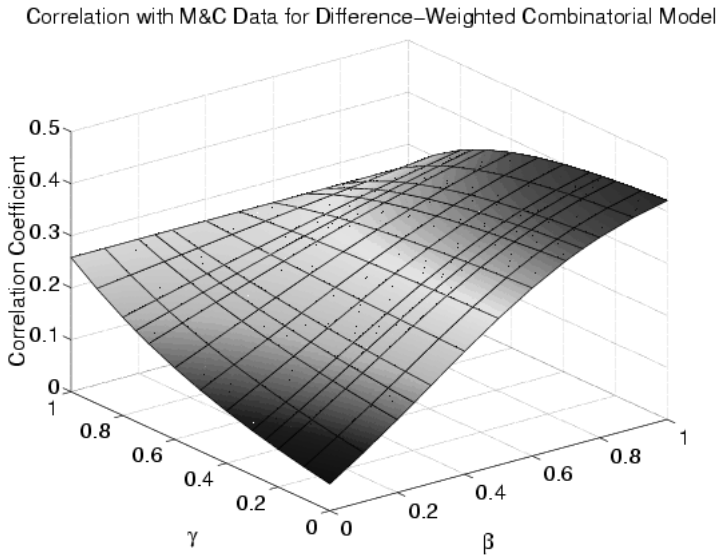

Figure 6.4: Correlation with M&C data for difference-weighted type-based model

synonymy judgements and negative correlation between the distance measures and the human synonymy judgements.

2. From the correlation values in Tables 6.2 and 6.3, we can see that the best performing measures at this task are the MI-based CRMs. If the first noun is considered to be the target noun, these perform best at high values of β. In other words, we get high correlation with synonymy judgements if the second noun has high precision retrieval of the first noun's co-occurrences, which is equivalent to the first noun having high recall retrieval of the second noun's co-occurrences.

3. Although being able to vary the parameters may give an unfair advantage to the CRMs over such a small data set, the smooth variation seen with the parameters, with consistently high performance at high values of β ($\beta > 0.5$ for the MI-based CRM), suggests that the improvement over the existing similarity measures is not due to the parameters giving rise to many different random permutations, or possible orderings, of the pairings.

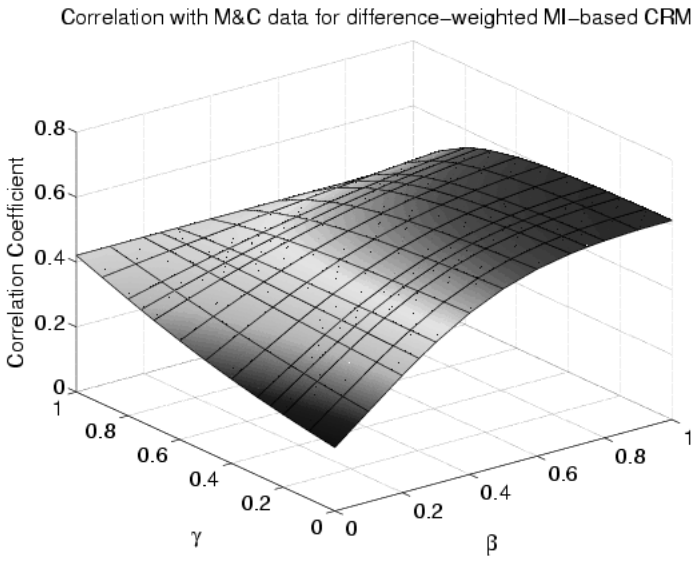

Figure 6.5: Correlation with M&C data for difference-weighted MI-based model

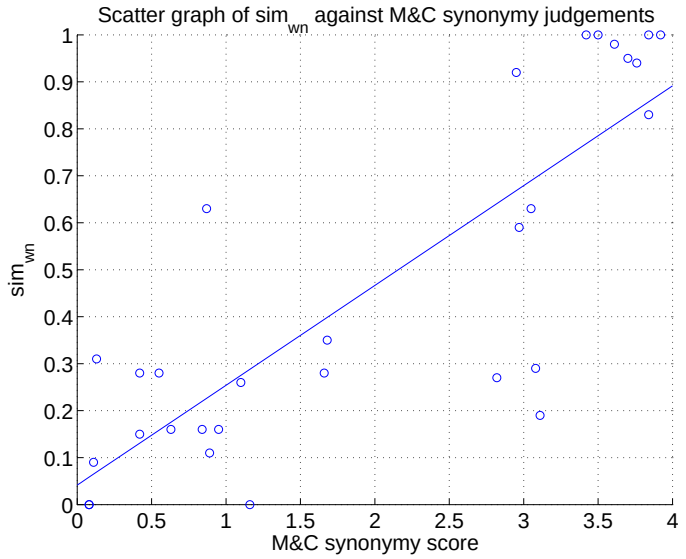

Figure 6.6: Scatter Graph for WordNet-Based Measure

6.1.3 Discussion

The question that remains is why the asymmetric settings in the CR framework perform much better than the symmetric settings, when synonymy apparently captures the symmetric concept of inter-substitutability of nouns. The answer may come from considering the relative frequencies of the nouns and/or the semantic relationship between the nouns in each pair in the study. The frequencies in the data and the precision and recall values (in the additive MI-based CRM) for the pairs are shown in Table 6.4.

Let us first consider some examples. In at least four of the pairs, the semantic relationship between the two nouns is one of hyponymy (*(food-fruit),(bird-cock),(bird-crane),(crane-implement)*). In the first three of these pairs, the hyperonym comes first and we might expect that this has some impact on the values of precision and recall. For example, in the pair (*food-*

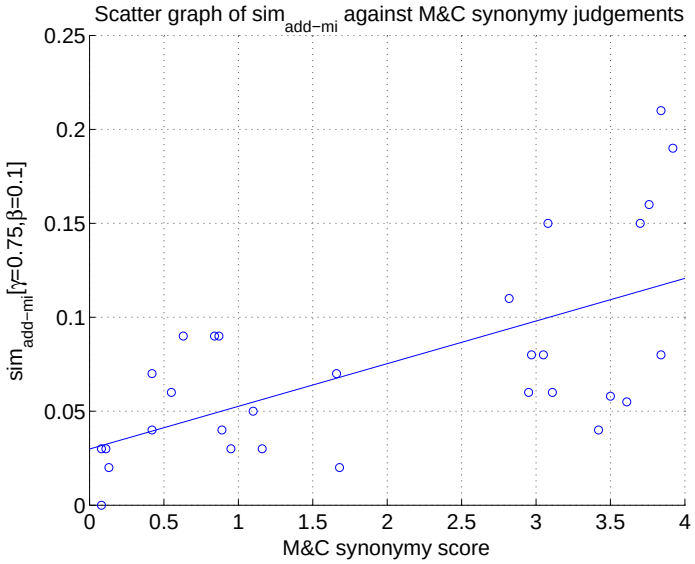

Figure 6.7: Scatter Graph for Additive MI-Based Model ($\gamma = 0.75, \beta = 1.0$)

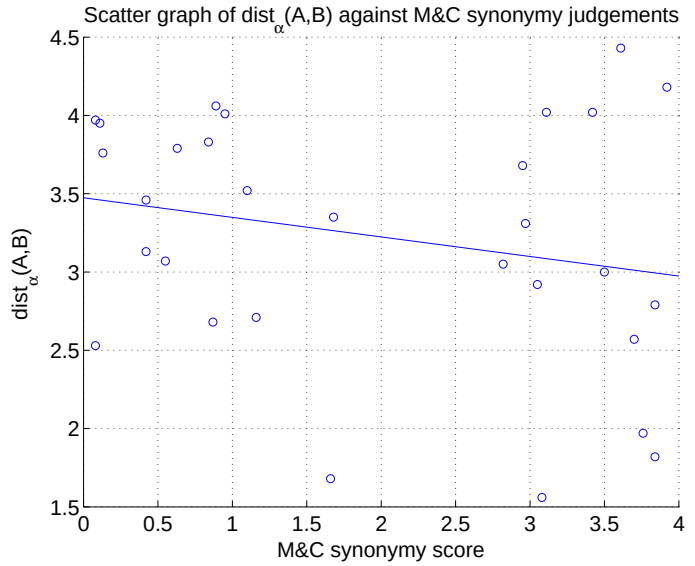

Figure 6.8: Scatter Graph for α-skew Divergence Measure ($dist_{\alpha}(A,B)$)

fruit), *fruit* is a hyponym of *food* and therefore we might expect to see that it has high precision retrieval of *food* since we would expect that all of the occurrences of *fruit* might be replaced by its hyperonym *food* but not vice versa.

However, this cannot be the explanation for the majority of pairs in which there is no obvious hyponymy relation. For example, in the pair *car-automobile* what is the difference between the apparent synonyms *automobile* and *car*, which leads to *automobile* having high precision retrieval ($\mathcal{P}_{mi}^{add}(automobile, car) = 0.7404$) of the co-occurrences of *car* but not high recall retrieval ($\mathcal{R}_{mi}^{add}(automobile, car) = 0.009$). The answer is that *car* is used much more frequently than *automobile* and therefore, even if both nouns can potentially occur in exactly the same contexts, *car* will be attested in more of these contexts. Accordingly, *car* is seen to have high recall retrieval of *automobile* and, conversely, *automobile* is seen to have high precision retrieval of *car*.

noun n_1	noun n_2	freq(n_1)	freq(n_2)	$\mathcal{R}_{mi}^{add}(n_2,n_1)$	$\mathcal{P}_{mi}^{add}(n_2,n_1)$	freq(n_1)> freq(n_2)	$\mathcal{P}(n_2,n_1) > \mathcal{R}(n_2,n_1)$
car	automobile	4146	11	0.009	0.7494	yes	yes
gem	jewel	49	106	0.170	0.106	no	no
journey	voyage	816	71	0.133	0.514	yes	yes
boy	lad	1525	198	0.093	0.396	yes	yes
coast	shore	227	142	0.147	0.289	yes	yes
asylum	madhouse	122	7	0.022	0.164	yes	yes
magician	wizard	22	39	0.134	0.081	no	no
midday	noon	8	15	0.107	0.058	no	no
furnace	stove	32	81	0.182	0.073	no	no
food	fruit	2476	691	0.147	0.309	yes	yes
bird	cock	908	69	0.028	0.241	yes	yes
bird	crane	908	42	0.014	0.275	yes	yes
tool	implement	698	30	0.016	0.180	yes	yes
brother	monk	1525	87	0.060	0.283	yes	yes
lad	brother	198	1081	0.290	0.086	no	no
crane	implement	42	30	0.045	0.029	**yes**	**no**
journey	car	816	4146	0.101	0.031	yes	yes
monk	oracle	87	53	0.053	0.082	yes	yes
cemetery	woodland	43	64	0.054	0.0358	no	no
food	rooster	2476	5	0.001	0.143	yes	yes
coast	hill	227	507	0.206	0.121	no	no
forest	graveyard	350	28	0.031	0.277	yes	yes
shore	woodland	142	64	0.010	0.173	yes	yes
monk	slave	87	124	0.116	0.092	no	no
coast	forest	227	350	0.142	0.099	no	no
lad	wizard	198	39	0.026	0.114	yes	yes
chord	smile	242	1375	0.064	0.026	no	no
glass	magician	2053	22	0.001	0.134	yes	yes
rooster	voyage	5	71	0.242	0.028	no	no
noon	string	15	641	0.0	0.0	no	no?

Table 6.4: Frequencies and precision and recall values (additive MI-based model) for the noun pairs in the M&C data

This frequency-related pattern is seen throughout the pairs in the study. For the majority of pairs (28 out of 30), $\mathcal{R}(n_1,n_2) > \mathcal{P}(n_1,n_2)$ when freq(n_1) > freq(n_2). There are two exceptions. One is *noon-string* for which both precision and recall are equal to zero. The other is *crane-implement*. The frequencies and precision-recall values for this pairing are fairly small and fairly similar. Further, the hyponymic term, *crane*, in this pair is more frequent than its hyperonym, *implement*. This is against the trend for other hyponym-hyperonym pairs. Accordingly, the fact that *implement* has higher recall retrieval of *crane* than *crane* of *implement* may not be surprising

even though *implement* is less frequent in occurrence than *crane*.

A similar result is observed when we consider the results for the α-skew divergence measure. The direction in which the α-skew divergence measure is smaller can be correctly predicted in 24 out of 30 cases by the relative frequencies of the nouns.

The fact that the majority (60%) the noun pairs are ordered in the direction of more frequent to less frequent may be the reason why an asymmetric measure is able to determine the second noun, in general, as having high precision retrieval of the first noun's co-occurrences. A measure which relies on recall, on this ordering of nouns, is unable to distinguish these pairs as being more semantically related than the unrelated pairs at the bottom of the list.

In order to test this hypothesis, we reordered the 40% of pairs that were ordered from less frequent to more frequent and reran the correlation experiment between $\mathcal{P}_{mi}^{add}(n_2, n_1)$ and the human synonymy judgements. However, surprisingly, the optimal similarity dropped slightly to 0.594 (at $\gamma = 0.7, \beta = 1.0$). It is not obvious why this is so, but it may have somthing to do with high distributional scores between pairs of words with low synonymy judgements. For example the noun *voyage* has a high recall retrieval of the co-occurrences of the semantically unrelated noun *rooster*. This is because there is very little co-occurrence data for *rooster* and it has fairly commonly occurring co-occurrences such as *have*, *make* and *like*. Although *have* is not normally considered a highly informative co-occurrence, it is in the context of *rooster* and *voyage* because these are both low frequency nouns.

In this section, we have considered only a small number of word pairs. In the next section, we perform a much larger study using rankings determined using the WordNet-based measure, which we have seen in this section to be highly correlated with the human synonymy judgements.

6.2 Comparison with WordNet

In this section we evaluate the ability of distributional similarity measures to predict semantic similarity by making comparisons with WordNet. An underlying assumption of this approach is that WordNet is a gold standard for semantic similarity, which, as already discussed, is not true. However, we believe that a distributional similarity measure which more closely predicts a semantic measure based on WordNet is more likely to be a good predictor of semantic similarity. Further, we have already seen in the previous section that Lin's (1997) WordNet-based measure is more closely correlated with human synonymy judgements than any of the distributional measures (Table 6.2) and that it performs well compared to other WordNet based measures (see Table 6.1). This suggests that Lin's WordNet-based measure will make a good basis for conducting a larger study of the distributional similarity measures' ability to predict semantic similarity than the synonymy judgements alone will allow.

In the first set of experiments we compare neighbour sets derived using the distributional similarity measures with those derived using the WordNet-based measure. We do this using the same technique as in Chapter 3, when we compared existing similarity measures, and as in Chapter 4, when we compared the CRMs with existing measures. In the second set of experiments we consider the different types of semantic similarity encompassed by the WordNet similarity measure. In particular, we examine the tendencies of different measures to select hyponyms or hyperonyms.

6.2.1 Neighbour Set Comparisons

As in previous chapters, we use an adaptation of the technique implemented by Lin (1998a). We compare the top 200 neighbours derived by each measure with the top 200 neighbours derived using the WordNet-based measure. As in earlier experiments (see Section 3.2.2), we use the neighbours computed for 1000 high frequency target nouns and 1000 low frequency target nouns from amongst the combined set of 2000 nouns. The measure used to compare two neighbour sets is a correlation or cosine coefficient between the rank scores in the neighbour sets:

$$C([w_k,...,w_1],[w'_k,...,w'_1]) = \frac{\sum_{w_i=w'_j} i \times j}{\sum_{i=1}^{k} i^2} \tag{6.1}$$

where each neighbour is given a rank score of $k - rank$ and where i and j are the rank scores of the nouns within each neighbour set.

Measure	Noun Frequency					
	high			low		
	parameters	C		parameters	C	
$dist_{L1}$		0.273	(0.090)		0.221	(0.113)
$dist_{L2}$		0.193	(0.073)		0.186	(0.084)
sim_{cos}		0.202	(0.091)		0.192	(0.103)
sim_{τ}		0.239	(0.063)		0.231	(1.073)
$dist_{js}$		0.288	(0.090)		0.245	(0.112)
$dist_{\alpha 1}$		0.290	(0.086)		0.270	(0.103)
$dist_{\alpha 2}$		0.264	(0.091)		0.174	(0.089)
sim_{cp}		0.252	(0.061)		0.224	(0.0067)
$sim_{cp_{rev}}$		0.218	(0.102)		0.191	(0.101)
sim_{jacc}		0.278	(0.077)		0.147	(0.086)
$sim_{jacc_{mi}}$		0.290	(0.087)		0.148	(0.090)
sim_{hind+}		0.297	(0.082)		0.264	(0.111)
sim_{lin}		0.307	(0.096)		0.210	(0.111)
sim^{add}_{type}	$\gamma = 0.25, \beta = 0.5$	0.299	(0.078)	$\gamma = 0.5, \beta = 0.4$	0.260	(0.091)
sim^{add}_{tok}	$\gamma = 0.4, \beta = 0.4$	0.281	(0.074)	$\gamma = 0.25, \beta = 0.1$	0.239	(0.077)
sim^{add}_{mi}	$\gamma = 0.25, \beta = 0.3$	**0.317**	**(0.094)**	$\gamma = 0.5, \beta = 0.1$	**0.274**	**(0.104)**
sim^{dw}_{type}	$\gamma = 0.0, \beta = 0.5$	0.283	(0.076)	$\gamma = 0.0, \beta = 0.0$	0.220	(0.105)
sim^{dw}_{tok}	n/a	0.273	(0.090)	n/a	0.221	(0.113)
sim^{dw}_{mi}	$\gamma = 0.0, \beta = 0.3$	**0.311**	**(0.094)**	$\gamma = 0.0, \beta = 0.0$	**0.264**	**(0.112)**

Table 6.5: Mean correlation between distributional neighbour sets and WordNet derived neighbour sets. Figures in brackets are standard deviations

We present the results in terms of the mean correlation coefficient for the 1000 high frequency target nouns and for the 1000 low frequency target nouns. Table 6.5 shows the results for the existing similarity measures discussed in Chapter 3 and for the optimal parameter settings in each CRM. As in Chapter 4, we would expect a mean overlap score of 0.08 by chance. Further, any difference between mean scores of greater than 0.035 is statistically "impossible" by chance, dif-

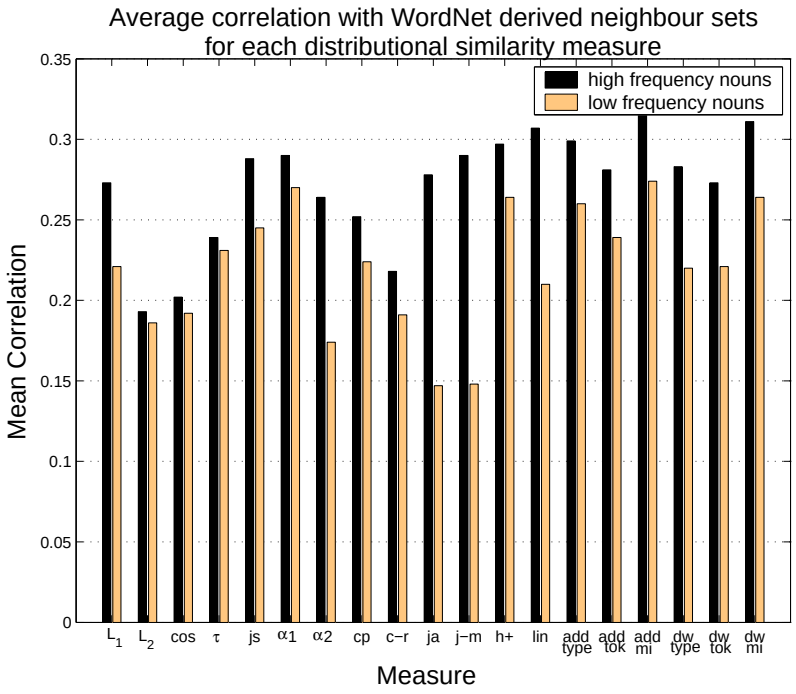

Figure 6.9: Barchart illustrating correlation with WordNet for each measure

ferences of more than 0.02 are significant at the 99% level (i.e. there is a less than 1% probability of such a difference occurring by chance) and differences of more than 0.01 are significant at the 90% level. The correlation values for each measure are also illustrated in Figure 6.9. From these results we can make the following observations:

1. The best performing CRMs are the MI-based ones. The additive MI-based CRM outperforms all other models and all of the existing measures for both high and low frequency nouns. The differences observed are statistically significant at the 90% level for all of the measures and models except the difference-weighted MI-based CRM.
2. The best performing existing measures are sim_{lin}, sim_{hind} and $dist_{\alpha 1}$.
3. All of the measures perform significantly better for high frequency nouns than for low frequency nouns.
4. Some of the measures (sim_{lin}, sim_{jacc} and $sim_{jacc_{mi}}$) perform considerably worse for low frequency nouns.

We now consider the effects of the two parameters in the CRMs on performance. Variation with respect to the parameters β and γ for each model is illustrated in Figures 6.10 to 6.14. From these figures, we can make the following observations:

1. For all models, and for both high and low frequency nouns, correlation with WordNet is higher for low values of β than for high values of β. In other words, neighbours according to the WordNet based measure tend to have high recall retrieval of the target noun's co-occurrences.
2. A high value of γ leads to high performance for high frequency nouns but poor performance for low frequency nouns. This suggests that the WordNet-derived neighbours of high frequency nouns also have high precision retrieval of the target noun's co-occurrences whereas the WordNet-derived neighbours of low frequency nouns do not.

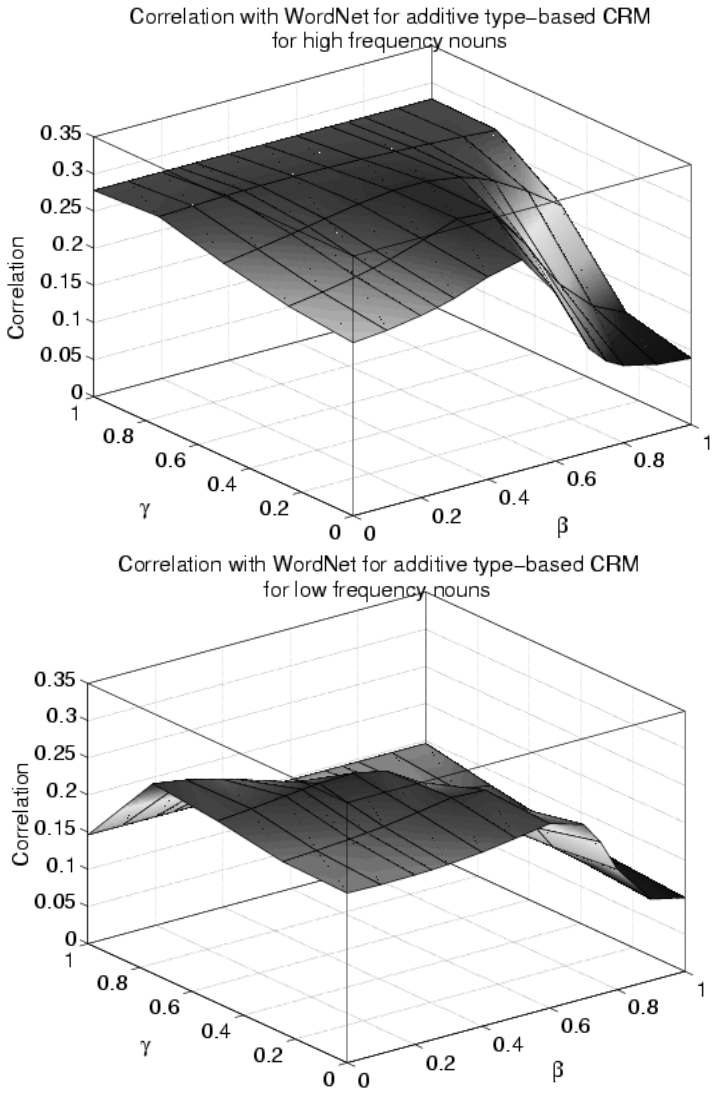

Figure 6.10: Variation in correlation with WordNet with respect to β and γ for the additive type-based CRM

This also explains why some of the existing measures (e.g. Jaccard's measure and Lin's measure) do very well for high frequency nouns but poorly for low frequency. We saw earlier that these measures are very similar to a $\gamma = 1$ setting in the CR framework and that they tend to pick out neighbours of similar frequency to the target noun i.e. high frequency neighbours for high frequency nouns and low frequency neighbours for low frequency nouns. The measures that do well for both frequency sets on this task are those that tend to select high frequency neighbours for both high frequency nouns and low frequency nouns.

Having considered the overall similarity between neighbour sets at different settings in the CR framework and WordNet-derived neighbours, we will now examine the tendencies for different parameter settings to select hyponyms or hyperonyms.

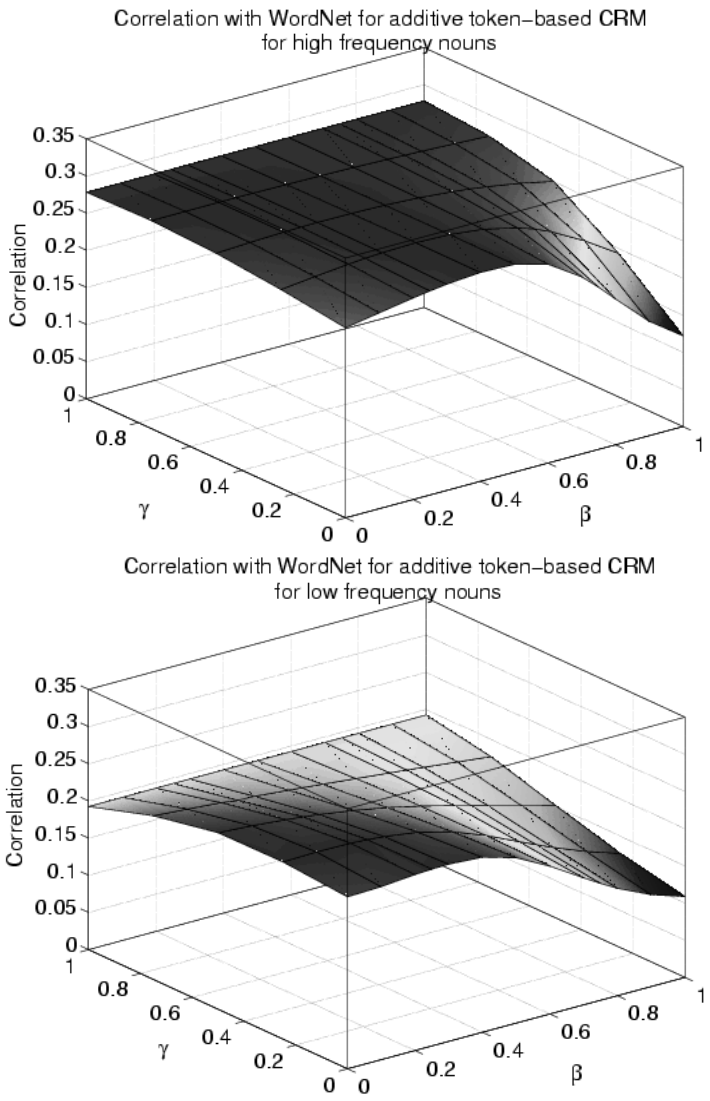

Figure 6.11: Variation in correlation with WordNet with respect to β and γ for the additive token-based CRM

6.2.2 Hyponyms vs. Hyperonyms

In this section we examine the relationship between direction in semantic similarity and direction in distributional similarity using two experiments. The implicit direction in semantic similarity is defined by the hyponymy relation. Synonyms are considered to be *equivalent*; and hyponyms and hyperonyms are moving away from the target word in degree of relatedness in different directions. In the first experiment we consider the values of similarity scores in both directions for asymmetric measures. In the second experiment, we consider how neighbour sets are made up in terms of hyponyms and hyperonyms.

6.2.2.1 Experiment 1

In this experiment we consider the relationship between precision and recall and between the values of $dist_{\alpha 1}$ and $dist_{\alpha 2}$ for pairs of hyponyms and hyperonyms. Our hypothesis is that the direction of hyponymy can be determined by the relative values of precision and recall. We expect that hyponyms will have higher precision than recall in their retrieval of a target noun's co-occurrences

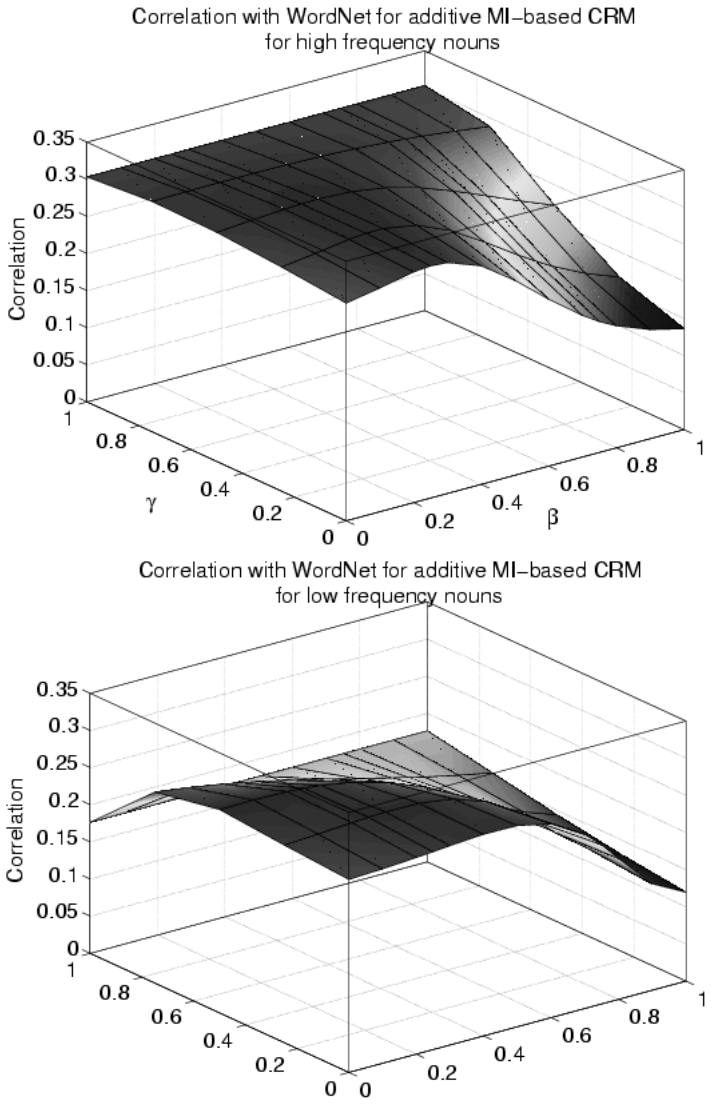

Figure 6.12: Variation in correlation with WordNet with respect to β and γ for the additive MI-based CRM

($\mathcal{P}(n_1,n_2) > \mathcal{R}(n_1,n_2)$ if n_1 is a hyponym of n_2) whereas hyperonyms will have higher recall than precision in their retrieval of the target noun's co-occurrences ($\mathcal{P}(n_1,n_2) < \mathcal{R}(n_1,n_2)$ if n_1 is a hyperonym of n_2). Similarly, for the α-skew divergence measure, we expect a hyperonym to be considered a better model for a target noun than that noun is model for the hyperonym ($dist_{\alpha 1}(n_1,n_2) < dist_{\alpha 2}(n_1,n_2)$ if n_1 is a hyperonym of n_2).

By considering every pair-wise combination of the 2000 target nouns, we extracted from WordNet a list of hyponyms and a list of hyperonyms for every target noun. Both direct and indirect relationships were included and every sense of every target noun in WordNet was considered. However, we excluded any hyperonyms or hyponyms from the lists where there were conflicts between senses. For example, sense 1 of noun 1 might be a hyperonym of sense 1 of noun 2 but sense 2 of noun 1 might be a synonym or hyponym of sense 2 of noun 2. In other words, a noun is never considered as both a hyperonym and a hyponym (or a synonym) of another noun.

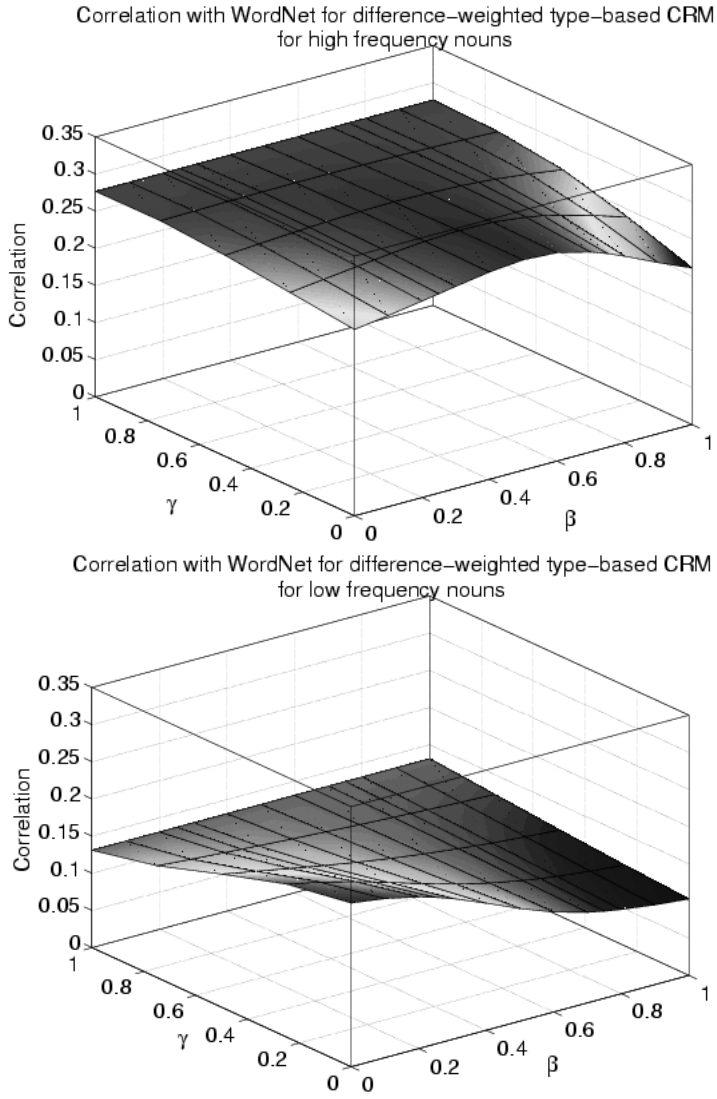

Figure 6.13: Variation in correlation with WordNet with respect to β and γ for the difference-weighted type-based CRM

After this process, we had 20,415 pairs of nouns (n_1, n_2) where n_2 is a hyponym of n_1. From this list, we constructed four smaller lists as follows:

hyperonyms of high frequency nouns This list contains 13,311 pairs (n_1, n_2) where n_1 is a high frequency noun and n_2 is a hyperonym of n_1.

hyponyms of high frequency nouns This list contains 18,308 pairs (n_1, n_2) where n_1 is a high frequency noun and n_2 is a hyponym of n_1.

hyperonyms of low frequency nouns This list contains 7104 pairs (n_1, n_2) where n_1 is a low frequency noun and n_2 is a hyperonym of n_1.

hyponyms of low frequency nouns This list contains 2107 pairs (n_1, n_2) where n_1 is a low frequency noun and n_2 is a hyponym of n_1.

From these descriptions we note that high frequency nouns have an average of 13 hyperonyms and 18 hyponyms and low frequency nouns have an average of 7 hyperonyms and 2 hyponyms.

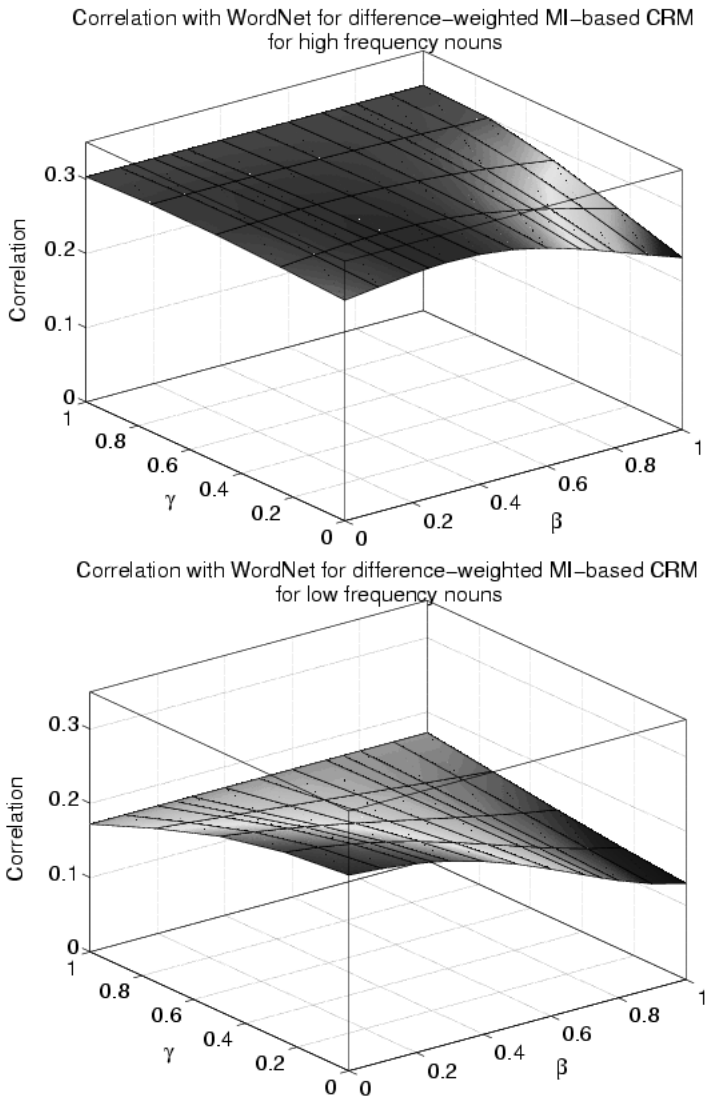

Figure 6.14: Variation in correlation with WordNet with respect to β and γ for the difference-weighted MI-based CRM

The fact that low frequency nouns have less of both type of relation may correspond with low frequency nouns having a smaller number of senses. More interesting is the fact that high frequency nouns tend to have more hyponyms than hyperonyms whereas low frequency nouns tend to have more hyperonyms than hyponyms. This would correspond with there being negative correlation between frequency and depth in the WordNet noun hierarchy. The scatter graph in Figure 6.15 shows the relationship between frequency of a noun in our data and average depth (averaged over all senses of the word) in WordNet. From this graph we can see that low frequency nouns appear to be distributed throughout the hierarchy and high frequency nouns tend to appear in the middle of the hierarchy. However, overall there is some negative correlation between depth and frequency. The graph shows the best-fit regression line and the value of Pearson's product-moment correlation coefficient for this data is -0.1761.

For each list and each CRM, we considered the precision and recall of the second noun's retrieval of the first noun's co-occurrences and computed the proportion where the inequality

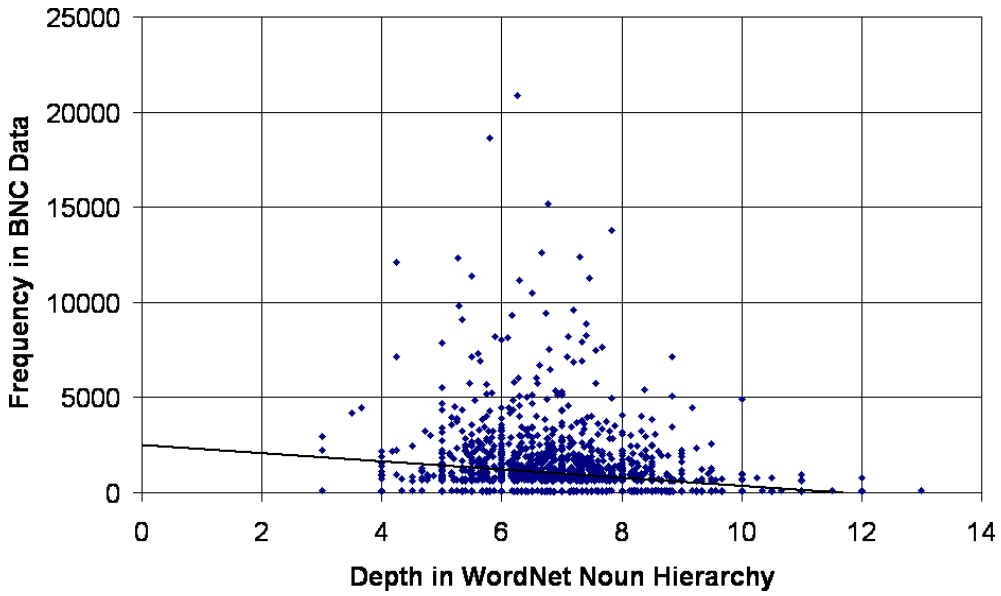

Figure 6.15: Frequency in BNC Data Against Depth in WordNet for 2000 Nouns

$\mathcal{R}(n_2,n_1) > \mathcal{P}(n_2,n_1)$ is true. A tie (or score of 0.5) is awarded when the two quantities are equal. Based on our hypotheses, we would expect this inequality to be true where n_2 is a hyperonym of n_1 and false when it is a hyponym of n_1. The results are summarised in Table 6.6. We do not include the difference-weighted token-based CRM since, in this model, precision always equals recall. In the case of the α-skew divergence measure, we consider the inequality $dist_{\alpha 1}(n_2,n_1) < dist_{\alpha 2}(n_2,n_1)$.

List	Model/Measure					
	sim^{add}_{type}	sim^{add}_{tok}	sim^{add}_{mi}	sim^{dw}_{type}	sim^{dw}_{mi}	$dist_{\alpha}$
hyperonyms of high frequency nouns	0.598	0.569	0.576	**0.632**	0.589	0.539
hyponyms of high frequency nouns	0.224	0.253	0.241	**0.210**	0.234	0.288
hyperonyms of low frequency nouns	**0.964**	0.942	0.955	0.942	0.959	0.928
hyponyms of low frequency nouns	0.707	0.717	0.723	**0.699**	0.704	0.715

Table 6.6: Proportion of pairs (n_1,n_2) where the recall of n_2's retrieval of n_1 is greater than its precision

From these results, we observe that a similar pattern is seen for all of the models and the α-skew divergence measure. Our hypothesis appears to correctly predict the hyperonyms of low frequency nouns and the hyponyms of high frequency nouns. However, it appears to incorrectly predict the hyponyms of low frequency nouns and predicts the hyperonyms of high frequency nouns with a success rate which is not much better than chance (0.5).

We can explain this by considering the relative frequencies of the word pairs. 55.5% of the hyperonyms of high frequency words are of a higher frequency than the target word (i.e. $freq(n_2) > freq(n_1)$), which contrasts to 26.4% of the hyponyms of high frequency words being of a higher frequency than the target word. 93.6% of the hyperonyms of low frequency words are

of a higher frequency than the target word, which contrasts to 71.5% of the hyponyms of low frequency words being of of a higher frequency than the target word. Accordingly, we could predict that a hyperonym is likely to be of a higher frequency than the target word and a hyponym is likely to be of a lower frequency than the target word. These results form a baseline for predicting the direction of hyponymy. The results for the CRM predictions and the α-skew divergence measure are fairly close to these baseline figures. We do note, however, that where the baseline does well (hyponyms of high frequency nouns and hyperonyms of low frequency nouns), the CRMs do slightly better; where the baseline does badly (hyperonyms of high frequency nouns and hyponyms of low frequency nouns), the CRMs do slightly less badly. In contrast, the α-skew divergence measure tends to do slightly worse than the baseline.

These results all concur with a situation where there is negative correlation between frequency and depth in WordNet. In this situation, the values of precision and recall are simply predicting which noun is of lower frequency rather than which is the hyponymic term. In general, it would seem that hyponyms are of lower frequency than their hyperonym. Picking the lower frequency noun as a hyponym will therefore tend to be correct. However, there will be some low frequency nouns at the top of the hierarchy and these will have higher frequency hyponyms. Therefore, for these pairs, the direction of the relationship may be incorrectly predicted. Further, this type of pair may constitute the bulk of the hyponyms of low frequency nouns list since (low frequency) nouns lower down the hierarchy will tend to have less hyponyms than (low frequency) nouns higher up the hierarchy. Accordingly, we more often incorrectly predict the hyponyms of low frequency nouns and the hyperonyms of high frequency nouns. In future work, we would like to attempt to disentangle these factors. For example, we could look at pairs of words of similar frequencies. However, the fact remains that relative frequency is a fairly good indicator of the direction of the hyponymy relation. If we take a weighted average over the results for each of the four lists of pairs, using just the relative frequency of the words, we can predict the direction of hyponymy in 70% of cases. This rises to 74% when the difference-weighted type-based CRM is used.

Having considered the relative values of precision and recall, we now consider the tendencies for hyperonyms and hyponyms to be amongst sets of neighbours derived using different measures. This means that the recall (or precision) of the retrieval by a potential neighbour is considered in relation to the recall (or precision) of other potential neighbours. This is in contrast to this experiment where we have considered the relative precision and recall values for a single potential neighbour.

6.2.2.2 Experiment 2

The hypothesis in this experiment is that a noun's hyperonyms will have greater recall retrieval of its co-occurrences than its hyponyms and conversely its hyponyms will have greater precision retrieval of its co-occurrences than its hyperonyms. Accordingly, we expect to find more of a noun's hyperonyms in neighbour sets generated using high recall measures than in neighbour sets generated using high precision measures. Similarly, we expect to find more of a noun's hyponyms in neighbour sets generated using high precision measures than in neighbour sets generated using high recall measures.

For this experiment, we used the same lists of pairs of nouns as in Experiment 1. For each CRM, we computed the proportion of n_2s found in neighbour sets of n_1 at different parameter

settings. We consider high precision neighbour sets ($\gamma = 0, \beta = 1$), high recall neighbour sets ($\gamma = 0, \beta = 0$) and high precision and recall neighbour sets ($\gamma = 1$). As in previous experiments, we considered neighbour sets of size $k = 200$. The results for each CRM are shown in Table 6.7. For comparison, results for three of the existing measures of distributional similarity and for the WordNet based measure are also shown.

		List of pairs (n_1, n_2) considered			
Measure		$freq(n_1)$=high		$freq(n_2)$=low	
Model	Parameters	n_1 ISA n_2	n_2 ISA n_1	n_1 ISA n_2	n_2 ISA n_1
additive type-based	$\beta = 0$	0.499	0.200	0.509	0.178
	$\beta = 1$	0.032	0.149	0.041	0.161
	$\gamma = 1$	0.440	0.233	0.100	0.206
additive token-based	$\beta = 0$	0.469	0.228	0.489	0.189
	$\beta = 1$	0.117	0.181	0.138	0.143
	$\gamma = 1$	0.410	0.259	0.277	0.198
additive MI-based	$\beta = 0$	**0.523**	0.254	**0.514**	0.221
	$\beta = 1$	0.087	0.204	0.074	0.170
	$\gamma = 1$	0.444	0.282	0.166	0.225
difference-weighted type-based	$\beta = 0$	0.509	0.202	0.348	0.221
	$\beta = 1$	0.213	0.236	0.050	0.169
	$\gamma = 1$	0.446	0.235	0.093	0.193
difference-weighted token-based	-	0.401	0.257	0.293	0.244
difference-weighted MI-based	$\beta = 0$	0.503	0.265	0.427	**0.252**
	$\beta = 1$	0.231	0.234	0.0866	0.184
	$\gamma = 1$	0.436	**0.286**	0.168	0.221
sim_{lin}		0.438	0.290	0.251	0.252
$dist_{\alpha 1}$		0.459	0.256	0.474	0.238
$dist_{\alpha 2}$		0.347	0.267	0.216	0.204
sim_{wn}		0.835	0.717	0.899	0.733

Table 6.7: Proportion of n_2s found in neighbour sets of size $k = 200$. We use the notation n_1 ISA n_2 to denote that the n_2s are hyperonyms of the n_1s.

The WordNet-based measure is obviously good at finding both hyperonyms and hyponyms for high and low frequency nouns. The fact that it does not get 100% suggests that some of the hyponym chains in WordNet are too long for all of the members of a chain to be within the 200 nearest neighbours of a target noun. The WordNet-based measure presents an upper bound on what it would be desirable for our distributional measures to find. The fact that it finds a greater proportion of the hyperonyms than the hyponyms is likely to be because a noun has less hyperonyms than hyponyms.

Of the existing distributional measures, the measure which performs the best across the board is $dist_{\alpha 1}$. It finds slightly less of the hyponyms than sim_{lin} but significantly more hyperonyms for high and low frequency nouns. The fact that it finds considerably more of the hyperonyms than the hyponyms may be related to its tendency to find higher frequency nouns as neighbours. This

would also explain why sim_{lin} is considerably better at finding hyperonyms for high frequency nouns than for low frequency nouns.

As in the previous experiment, we see a similar pattern of results for all of the CRMs with respect to β and γ. All of the models are best at finding hyperonyms at low values of β i.e. neighbours which have high recall retrieval of the target noun's co-occurrences are more likely to be hyperonyms. This is true for both high frequency nouns and for low frequency nouns. Performance at $\gamma = 1$ is similar to $\beta = 0$ for high frequency nouns and to $\beta = 1$ for low frequency nouns.

However, with regards to hyponyms, the results are more surprising. A greater proportion of hyponyms are found in high recall neighbour sets than in high precision neighbour sets. However, the greatest proportions of hyponyms, for high and low frequency nouns, are found amongst the high recall and high precision neighbours of the $\gamma = 1$ setting. This suggests that the high precision neighbours do not tend to be either hyperonyms or hyponyms. However, there is a much stronger tendency for high precision neighbours not to be hyperonyms. The high precision measures find a lot fewer hyperonyms than they do hyponyms. This, therefore, does support the idea that high precision neighbours tend to be less general nouns than high recall neighbours. The fact that they are not hyponyms suggests that they are more likely to be siblings in the WordNet hierarchy such as *apple* and *pear*. This contrasts to the high recall measures, which seem more likely to include *fruit* as a neighbour of *apple*.

However, the performance is generally fairly poor. The best performing measures, the MI-based models, still only find approximately 50% of a noun's hyperonyms and 25% of a noun's hyponyms. These proportions are considerably less than for the WordNet based measure (85% and 70% respectively). One explanation is that these results could be skewed by hyperonyms and hyponyms of obscure senses in WordNet that are unattested in our corpus.

6.2.2.3 Discussion

We have seen that a hyperonym does not necessarily have higher recall retrieval of its hyponyms co-occurrences than it has precision retrieval those co-occurrences, other than a correspondence which is most likely to be due to a correlation between noun frequency and depth in WordNet.

However, we have seen that if we compare precision and recall scores of different neighbours of the same target noun, high recall neighbours include more hyperonyms than hyponyms and high precision neighbours include more hyponyms than hyperonyms. This result is constant for both high and low frequency nouns.

So far in this chapter, we have looked at the ability of distributional similarity measures to predict human synonymy judgements, to generate sets of neighbours akin to those derived from WordNet and to predict hyponymy relationships between pairs of semantically related words. In the next section we consider their performance on an application-based semantic task.

6.3 Spelling Correction Task

In this section we examine the ability of distributional similarity measures to predict semantic similarity using a spelling correction task. We adopt a variant of the task that was employed by Budanitsky and Hirst (2001) to evaluate five WordNet based measures. The task exploits the property of lexical cohesion, that is, texts tend to contain many semantically related words. We

will first discuss the motivation behind the task and the methodology used. We will then present our results for a number of the distributional similarity measures and our conclusions.

6.3.1 Motivation

The use of an application-based task allows us to evaluate semantic similarity independent of any gold standard. The task relies on the properties of real text and not the intuitions of human judges. The property exploited is that texts tend to be cohesive, that is, they contain many semantically related words (Morris & Hirst, 1991). In other words, the occurrence of one word makes the occurrence of other semantically similar words in the locality more likely (Church, 2000).

This property was exploited in the real-word spelling error corrector described in (Budanitsky & Hirst, 2001). A real-word spelling error, or malapropism, is typographically or phonologically similar to the intended word and is produced by a writer mistyping the word (e.g. *dairy-diary*) or being confused between the two words (e.g. *principle-principal*). In these experiments, the spelling variations of a word *w* are defined as those words in the dictionary that can be derived from *w* by a single insertion, deletion or transposition. Real-word spelling errors cannot be detected by naïve spelling correctors since the incorrectly spelt word exists in the dictionary. However, Budanitsky and Hirst (2001) propose a spelling correction method which is based on the following assumptions:

1. A real-word spelling error is unlikely to be semantically related to the text.
2. Frequently, the writer's intended word will be semantically related to nearby words.
3. It is unlikely that an intended word that is semantically unrelated to all those nearby will have a spelling variation that is related.

As Budanitsky and Hirst discuss, the first assumption is actually untrue for human-generated spelling errors. However, it is true for the spelling errors created by their malapropism generator and it is also true in our experiments. We now discuss the methodology used in more detail.

6.3.2 Methodology

We will first describe the methodology used by Budanitsky and Hirst (2001) to evaluate five WordNet based measures. We will then discuss the modifications we have made.

Budanitsky and Hirst (2001) took 500 articles from the Wall Street Journal and, after removing proper nouns and stop-words, replaced one word in every 200 with a spelling variation, choosing always a WordNet noun with at least one spelling variation. This gave them a corpus with 107, 233 such words, 1408 of which were malapropisms. They then attempt to correct the spelling errors using a two-step process. In the first step, a word is *suspected* of being a spelling error if it is not semantically related to any word within a certain scope. In order to make the Boolean-valued "is semantically related or not" decision, they consider whether the similarity between the two words is above a certain threshold, which is set in advance at a value which can separate highly related pairs of words from unrelated pairs as determined in the study by Rubenstein and Goodenough (1965). In the second step, a suspected word is *detected* as being a spelling error if it has a spelling variation that is semantically related to any word within the scope. Budanitsky and Hirst (2001) vary the amount of search scope considered from the single paragraph, to one paragraph on either

side, to two paragraphs on either side, to the whole document. For each measure and scope, they measure precision and recall of errors at both the suspicion and detection steps.

We constructed a variant of this task as follows:

- First, we used the BNC as our data-set.
- Second, rather, than replacing every *n*th word, which meets certain conditions, in advance, we generated the errors on the fly. This meant, theoretically, that we could generate a higher proportion of spelling errors than Budanitsky and Hirst. Our errors do not permanently affect the text and therefore we can be sure that there are no other spelling errors within the search scope without limiting the number of malapropisms generated. However, in practice, we generated the same proportion of errors as Budanitsky and Hirst by giving our random generator the same prior probability of generating an error as in those earlier experiments (0.0131). The reason for this was that the prior probability of a spelling error affects the baseline considerably. For example, we could imagine a spelling error detector which simply guesses whether a word is spelt correctly or not. The precision of such a detector will equal the prior probability of a spelling error. The recall of such a detector will increase as the probability of guessing a word is a spelling error increases. Recall will be maximum, at 1, when the spelling detector guesses that every word is a spelling error. The maximum F-score in this case is given by:

  $$\frac{2 \times P(\text{spelling error})}{1 + P(\text{spelling error})} \tag{6.2}$$

  Accordingly, the baseline F-score increases as the probability of a spelling error increases. When this probability is 0.0131, the baseline F-score is 0.0259. However, if we increase the probability to 0.5 then the baseline F-score will increase to 0.666.
- The third modification we made was to consider scope in terms of a window of size N on either side of the target word. This was largely so that we could discard paragraph boundaries in our pre-processing of the BNC. We would expect this to have a negative effect on performance since a paragraph is a natural unit of text whereas our windows are not.
- Fourth, we did not set the threshold for semantic relatedness in advance. Instead, we varied *k*, the number of nouns considered as semantically related rather than the similarity threshold. One reason for this was that many of the distributional similarity measures were unable to split the semantically related and semantically unrelated pairs at any value. It also makes the experimental conditions in this experiment consistent with those in the pseudo-disambiguation experiment. We also intended to perform the same variation as in the pseudo-disambiguation experiments, where the similarity threshold itself is varied. However, our results in Section 6.3.3 suggest that there would be little to be gained by this.

  We note, however, that in these experiments, a word may be considered to be a neighbour of itself. This contrasts with the pseudo-disambiguation experiments. In the pseudo-disambiguation experiments, we could not consider a word as a neighbour of itself since we knew in advance with which verb the target word went more frequently. However, in these experiments, we have not constructed the test set in such a way that we have any prior knowledge about other words close in physical proximity to the target word. The presence of the target word itself, or its spelling variant, within the search scope would seem to be a good indicator of the intended word and there is no reason to exclude it as a semantically or distributionally related word. However, as our results show, the presence of the word itself is such a good indicator of which spelling variation was intended that we are left doubting whether other semantically, or distributionally, similar words have any additional effect.

As in previous experiments, we only consider similarities between 2000 target nouns. Accordingly, we only consider the presence of other nouns also within the set of 2000 target nouns and possible spelling variations which are in the set of 2000 target nouns.

- Finally, because we are optimising parameters, we divided the test instances into a development set and a test set. The development set constitutes 60% of the test instances and is used to find the parameter settings that optimise the F-score of the two-stage detection process. The test set constitutes the remaining 40% of the test instances and we use it to measure precision, recall and F-score of detection at the optimised parameter settings.

6.3.3 Results

In this section we present our results on the spelling correction task. Our hypothesis is that if distributional similarity measures are good predictors of semantic similarity, then they will perform the task as well a semantic similarity measure.

We first consider the parameter settings which optimise F, the harmonic mean of precision and recall of the detection process, in the development set using the measure of semantic relatedness, sim_{wn}. Graphs of detection F against k and scope are shown for sim_{wn} (Figure 6.16).

From these results, we observe that optimal performance of 0.31 for high frequency nouns and 0.22 for low frequency nouns is achieved when $k = 0$, i.e. the only neighbour considered is the target word itself. Performance drops rapidly when semantically related words are also considered. When a single neighbour is considered, detection F drops to 0.22 for high frequency nouns and 0.2 for low frequency nouns. When the ten nearest neighbours are considered, detection F is 0.1 for high frequency nouns and 0.08 for low frequency nouns. Our figures are not directly comparable with the those quoted by Budanitsky and Hirst (2001) due to the differences in methodology. However, it is interesting to note that Budanitsky and Hirst (2001) achieve an optimal detection F of 0.2 for Lin's measure. Since every noun in WordNet is considered a potential neighbour in their experiments, we might expect them to do better than us. However, we also note that their use of a fixed similarity threshold means that they do not generally consider the case where the only neighbour of a word is itself.

Figure 6.17 shows what is happening in terms of precision and recall as k varies at scope = 150. The reason for the drop in the value of the F-score is that considering semantically related words as well as the word itself dramatically reduces precision. In other words, the presence of even the most semantically related word does not appear to be a good indicator that the word is correctly spelt, at the suspicion stage, or that the spelling variation is the intended word, at the detection stage.

Accordingly, it seems that knowledge of semantically related words does not help in this task. Only at low scopes, scope = 10, when we would be unlikely to see a word repeated, do we see any benefit of considering the word's neighbours. However, rather than considering a word's semantic neighbours at this narrow scopes, it seems much more advantageous to look for a repetition of the word itself in a wider search scope.

As the results for the semantic measure were not what we expected, we only considered performance for two of the distributional measures. Figure 6.18 shows how the F-Score varies with k and scope when sim_{lin} is used.

Unsurprisingly, the pattern of results for sim_{lin} is the same as for sim_{wn}. This pattern of results

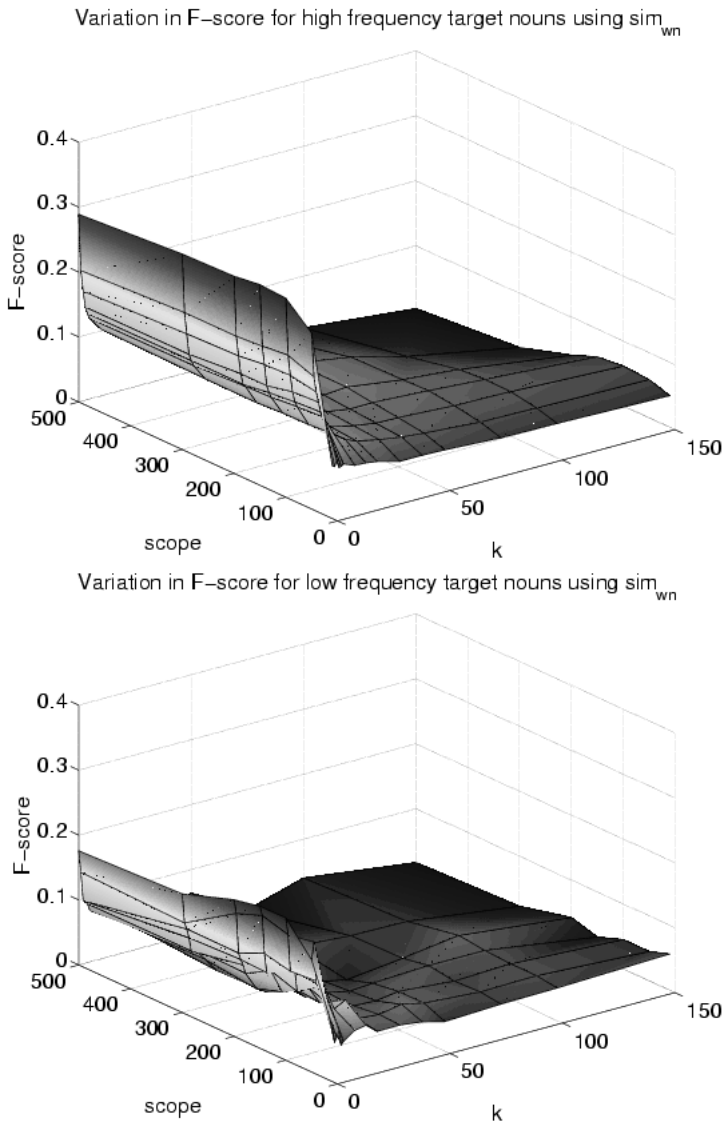

Figure 6.16: Variation in F on spelling correction task with respect to parameters k and scope for sim_{wn}

is also repeated when we use the additive MI-based CRM, which has been previously the best performing CRM in our experiments. Using this model, we found that performance did not drop so fast as the number of neighbours considered was increased if the neighbours had fairly high precision retrieval of the target nouns co-occurrences. Figure 6.19 shows how performance varies with k and scope at ($\gamma = 0, \beta = 0.9$). Figure 6.20 shows how performance varies with γ and β at $k = 1$ and scope=150.

Optimal performance with the additive MI-based CRM at $k = 1$ and scope=150 for high frequency target nouns is 0.30. This is only slightly less than at $k = 0$ (0.31) and better than at $k = 1$ for sim_{wn} (0.21) and sim_{lin} (0.25). We might think that this makes this CRM a better similarity measure than sim_{wn} and sim_{lin} for use in a spelling correction task. However, we note that the most likely reason for high precision neighbours reducing performance the least is that high precision neighbours tend to be low frequency words i.e. ones that are unlikely to occur. Consequently, considering whether the most precise neighbour is present is less likely to change the result from

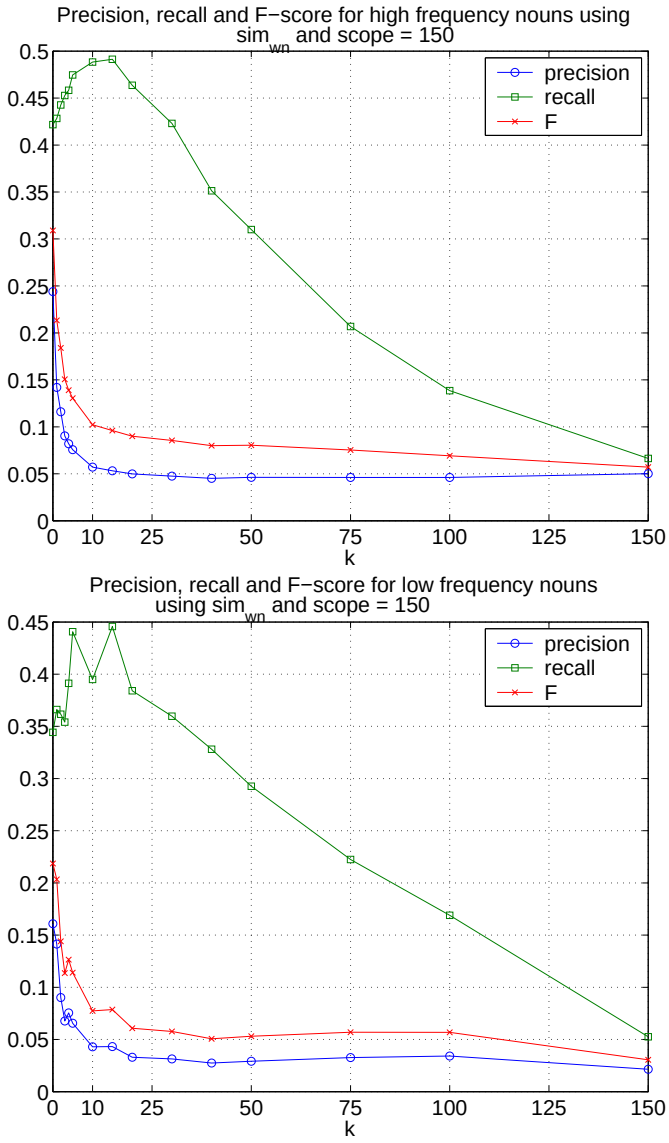

Figure 6.17: Variation in precision, recall and F as the number of nearest neighbours considered, k, is varied, for sim_{wn} at scope = 150 words

when only considering whether the target word is present. For us to be convinced that searching for highly precise neighbours is useful in the spelling correction task, we would need to see some increase in performance over that at $k = 0$.

6.3.4 Discussion

There are a number of possible reasons why the results for $k > 0$ are so much worse than for $k = 0$ on this task. One is that the hypothesis that texts contain semantically related words is wrong, but this would not fit with our intuitions about text cohesiveness. Second, we have implemented scope in terms of a window of words rather than paragraphs, which means that we cross natural text boundaries in our search for semantically related words. However, if this was the problem, we should also get worse performance when we search for the word itself. Third, we are only considering 2000 nouns as potential neighbours. However, from our earlier results on other tasks,

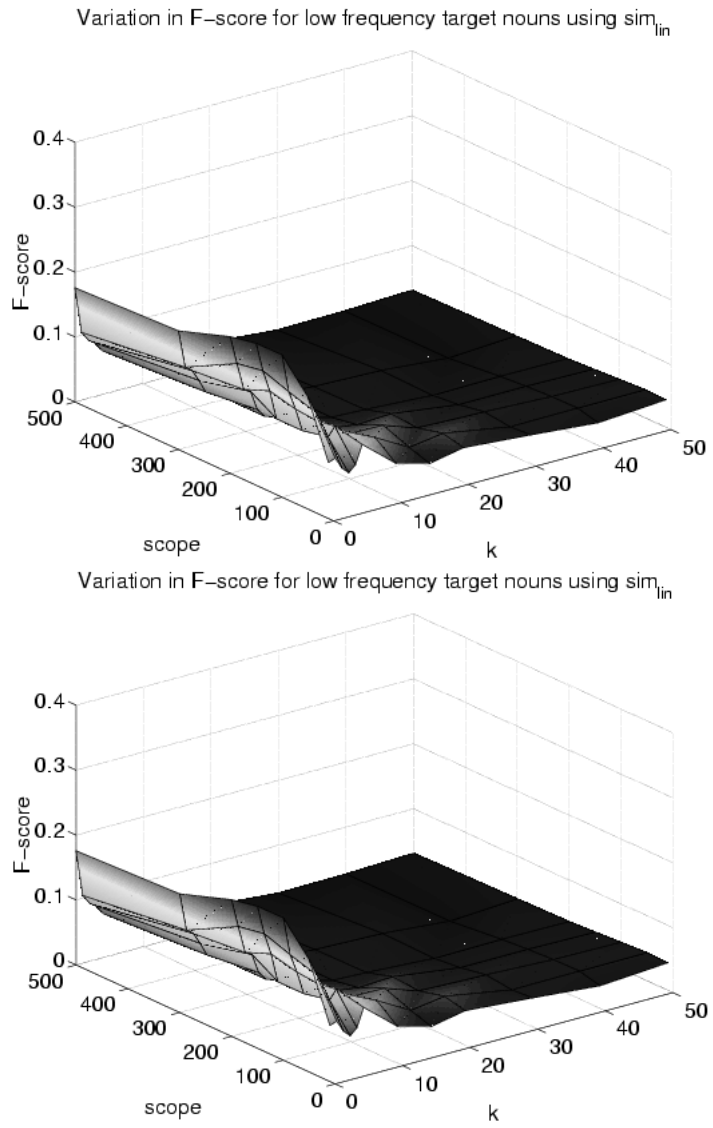

Figure 6.18: Variation in F on spelling correction task with respect to parameters *k* and scope for sim_{lin}

we would not expect this to adversely affect the performance to this extent. Fourth, the type of semantic relatedness considered is based on WordNet and second order affinities. The semantically related words which occur in a naturally cohesive text do not have to be lexically substitutable. By the very nature of the task, the related words occurring in the text will have first order affinities with the target word. Thus, in using WordNet or distributional similarity measures to carry out this task, we are attempting to predict first order affinities from second order affinities. Our results show that words that have second order affinities and/or are related by a taxonomical relation, do not also have strong first order affinities.

The results from this task are such that it is difficult to draw any conclusions about the ability of distributional similarity measures to predict semantic similarity. We have seen that the semantic measure does not increase the performance over simply searching for occurrences of the target word and its spelling variations within the search scope. Accordingly, what does it mean if a different measure increases or decreases the drop in performance? We may think that a smaller

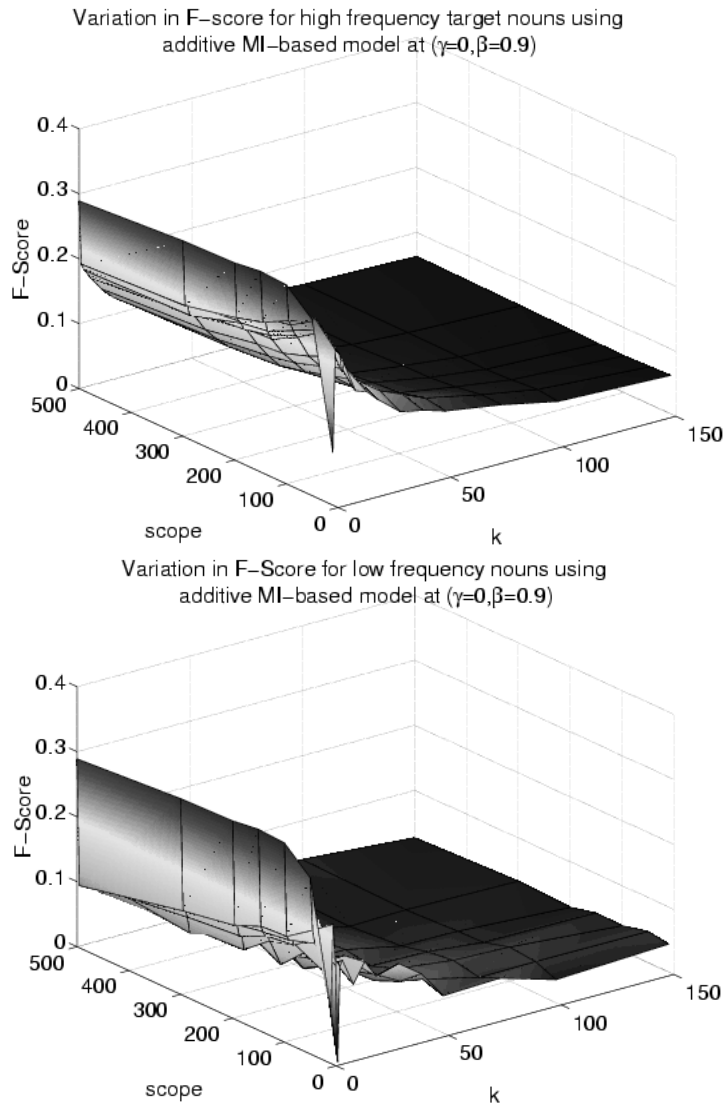

Figure 6.19: Variation in F-Score on spelling correction task with respect to parameters *k* and scope for the additive MI-based CRM at ($\gamma = 0, \beta = 0.9$).

drop in performance means that we are on the right track to finding words that will help us correct spelling errors. However, whether these are words which are related by WordNet-type linguistic relations is another matter. In order to do better on the task it would seem likely that we need to consider a larger number of words (possibly of other parts of speech) and a larger number of semantic relations.

6.4 Conclusions

In this chapter we have examined the ability of distributional similarity measures to predict semantic similarity. We have seen that there is positive correlation between distributional similarity values and human synonymy judgements but it is not as high as between the semantic, WordNet-based, measure and the human synonymy judgements.

We then evaluated the ability of distributional similarity measures to predict semantic similarity as defined by a measure based on the WordNet noun hyponym hierarchy. Using this task, we

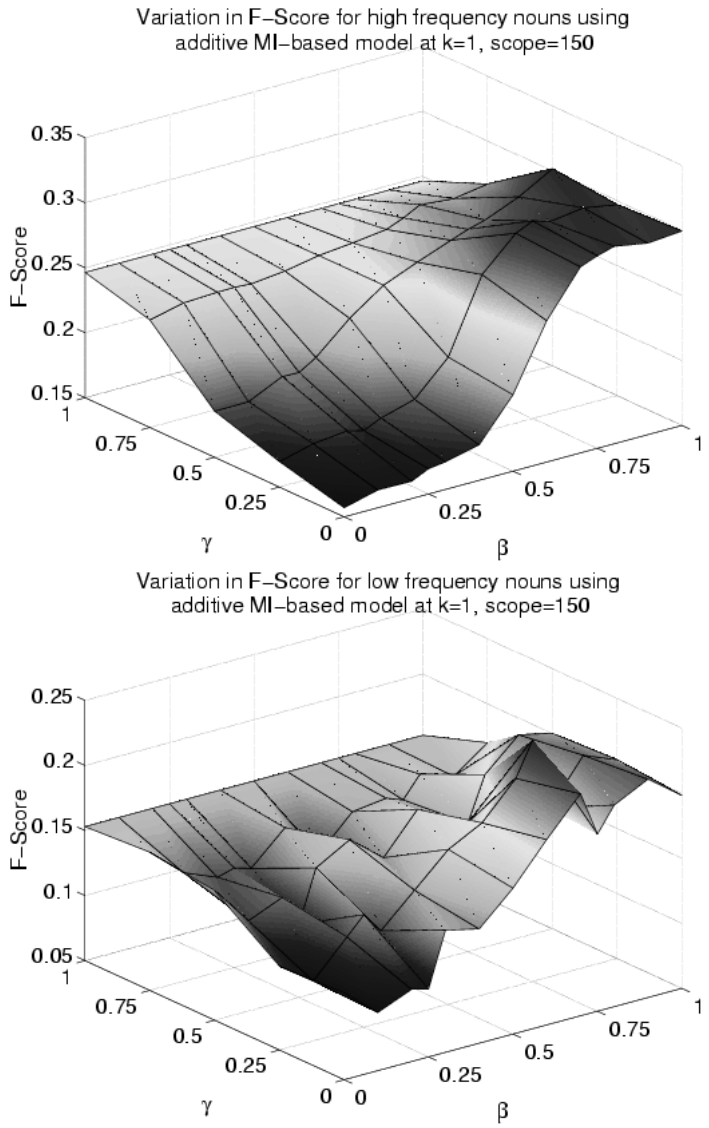

Figure 6.20: Variation in F-Score on spelling correction task with respect to parameters β and γ for the additive MI-based CRM at $k = 1$ and scope=150.

can establish the properties of distributional similarity measures which might be desirable in semantic applications. First, we have seen that the best-performing distributional measures were the MI-based CRMs. This indicates that the most selective verbs are the most salient in the context of predicting semantic similarity. Second, the difference in extent to which each word occurs in each context does not appear to be a major factor. As in the language modelling application, it appears that it is sufficient to know that each word has occurred in that context. However, we saw that for all CRMs, and for both high and low frequency target nouns, correlation with WordNet derived neighbours is higher for low values of β than for high values of β. In other words, neighbours according to the WordNet-based measure tend to have high recall retrieval of the target noun's co-occurrences. A high value of γ leads to high performance for high frequency target nouns but poor performance for low frequency target nouns. This suggests that WordNet-derived neighbours of high frequency target nouns also have high precision retrieval of the target noun's co-occurrences whereas the WordNet-derived neighbours of low frequency target nouns do not. It also explains

why measures which are closely simulated by a high value of γ in the CR framework do well for high frequency nouns but not for low frequency nouns. The measures which do well at this task are those that tend to select high frequency neighbours for both high frequency target nouns and low frequency target nouns.

We have also seen that there appears to be a relationship between precision-recall in the CR framework and the direction of the semantic relationship between the pairs of words. In particular, high recall neighbours tend to include more hyperonyms than hyponyms and high precision neighbours tend to include more hyponyms than hyperonyms. The reason for this relationship between precision-recall and hyponymy might be that hyperonyms are more general and thus tend to occur in a greater number of distinct contexts than their hyponyms. However, there is also a strong correlation between the number of distinct contexts in which a word appears and its total frequency. Further, a very frequent word is more likely to have been seen in more of its possible contexts and therefore is likely to appear to have high recall of a related word even if there is no hyponymy relation between the word. However, we have seen in a pair of nouns where frequencies are approximately equal, the precision-recall values were able to pick out the hyponym independent of the relative frequencies. This is an area for further study.

Finally, we attempted to use a semantic application-based task to test the measures' ability to predict semantic similarity independent of human judgements. The related words that occur are ones which have first order affinities with the target noun, by the nature of the fact that they occur in the same document. However, we have seen that, for the set of words considered, the semantic similarity measure was unable to produce any increase in performance over that obtained by just considering the word itself. This suggests that second order affinities are not useful in predicting first order affinities. Consequently, it is difficult to draw conclusions from this task about the distributional similarity measures considered.

Chapter 7

Conclusions and Future Directions

This thesis has been a study of measures of lexical distributional similarity and their applications. In this chapter, we summarise our major conclusions, present our major contributions and finish with the main directions for further work.

7.1 Summary of Conclusions

In this section, we present a summary of the conclusions made in each chapter of the thesis and highlight some areas for further work.

7.1.1 Chapter 1

In Chapter 1, we outlined our major hypotheses. First, similarity is an inherently asymmetric concept. Thus, a distributional similarity measure that can exploit this will be useful in more applications than an unparameterised or symmetric measure. Second, in the measurement of similarity, it is sufficient to know that two words can both occur in a particular context. Occurring in more of the same distinct contexts is more important than occurring in more of the same indistinct contexts. Third, measures that give more weight to more informative contexts will perform better than measures that give more weight to more frequently occurring contexts.

7.1.2 Chapter 2

We started Chapter 2 by considering what it means for two words to be distributionally similar. A fairly loose definition of lexical distributional similarity is that two words are distributionally similar if they appear in similar contexts. This corresponds with what Grefenstette (1994a) termed a second order affinity between words. Further, by defining context in terms of grammatical relations between words, distributional similarity can be thought of in terms of lexical substitutability (Church et al., 1994). Viewed in this way, distributional similarity is an inherently asymmetric concept.

We then discussed fifteen potential applications of knowing distributionally similar or lexically substitutable words. Some of these applications take advantage of the much-discussed link between lexical distributional similarity and semantic relatedness: if two words mean similar things

then they will be used in similar ways. Other applications, such as probability estimation techniques, make use of the property of distributional similarity directly: if two words tend to occur in similar contexts then we can predict the contexts in which one occurs from the contexts in which the other occurs.

The final section in Chapter 2 looked at possible approaches to evaluation in this area. We argued that humans are unable to judge lexical distributional similarity independent of semantic similarity and that, in any case, there is no single right answer as to which words are most distributionally similar. The best measure of distributional similarity will be the one which returns the most useful neighbours in the context of a particular application and thus leads to the best performance in that application.

7.1.3 Chapter 3

In Chapter 3 we introduced fifteen existing measures that have been previously proposed or adopted for measuring lexical distributional similarity. We compared the neighbours that these measures produce for 2000 nouns from the same direct-object data extracted from the BNC and examined their tendencies to select high or low frequency words as neighbours. We saw that Jaccard's Coefficient and the Dice Coefficient produce identical neighbour sets as do the cosine measure and Pearson's product-moment correlation coefficient, but otherwise there is a large variety in the neighbour sets produced by the different measures. However, the measures can be classified in terms of their tendencies to select words of particular frequencies (or distribution width) as neighbours. The majority of the measures considered tend to select words of a higher frequency than the target noun as neighbours. Two measures (sim_{lin} and sim_{jacc}) tend to select words of a similar frequency to the target noun as neighbours and one of the measures ($sim_{cp_{rev}}$) tends to select words of a lower frequency than the target noun as neighbours.

Finally, we discussed how knowledge of the tendencies of measures to select words of particular frequencies might impact on different applications. First, it will obviously have computational benefits. For example, if a measure always selects its neighbours from the set of most frequently occurring words, performance on a task will not be improved by considering more than these words as potential neighbours. Second, knowledge of the frequency characteristics of a measure might also influence the choice of measure in a particular application. For example, a distributional similarity measure that tends to select higher frequency words as neighbours would be more appropriate in a text simplification task.

7.1.4 Chapter 4

In Chapter 4, we defined a framework for measuring lexical distributional similarity based on the concept of substitutability. We cast the problem as one of *co-occurrence retrieval* (CR) for which we can measure precision and recall by analogy with the way that they are measured in *document retrieval*.

Since similarity in our framework is defined by substitutability, it is inherently asymmetric. However, the CR framework does not require similarity to be viewed as asymmetric since there are settings in each model ($(\gamma = 1, \beta)$ and $(\gamma, \beta = 0.5)$) at which it is symmetric and there is a model (the difference-weighted token-based model) which is always symmetric. These symmetric

settings may be thought of as capturing our notion of *inter-substitutability*.

This idea of asymmetry in substitutability makes CR comparable with measures based on relative entropy or the confusion probability. Since $\mathcal{R}(n_1,n_2) = \mathcal{P}(n_2,n_1)$, we could consider using $D(n_1||n_2)$ and $D(n_2||n_1)$ instead of precision and recall. However, our framework has the advantages that it directly defines a similarity measure, does not require smoothing of the base language model and allows us to explore systematically other questions we have about the relative importance of different features. In order to do this, we defined six different co-occurrence retrieval models (CRMs). Three of the models are additive models and the other three are difference-weighted models. This allows us to explore the effect of the difference in feature values for shared features. For both additive and difference-weighted models, we have defined type-based, token-based and MI-based models. This allows us to explore the effect of weighting different features by some degree of relevance or association.

The remainder of Chapter 4 was devoted to analysing the neighbour sets produced by the CR framework and determining the CR characteristics of existing distributional similarity measures. We saw that, in general, high recall neighbours tend to be high frequency nouns, high precision neighbours tend to be low frequency nouns and a high harmonic mean tends to produce neighbours of a similar frequency to the original noun. However, the CRM used also has an effect on the frequency of the neighbours produced. In particular, the difference-weighted models tend to be flatter with respect to the parameters β and γ and produce neighbours of a similar frequency to the target noun. Also of note is the ($\gamma = 0, \beta = 0.5$) setting in the additive MI-based model which appears to produce neighbour sets with no apparent biases to either high or low frequency nouns.

With regard to existing distributional similarity measures, we saw that six of those considered can be directly derived or very closely approximated by the CR framework. A summary of these findings was presented in Table 4.9. Knowledge of the CR characteristics of existing measures helps us to understand these measures. For example, high recall measures such as sim_{τ}, sim_{hind}, $sim_{\alpha 1}$ and sim_{cp} tend to select higher frequency nouns. The measures that require less recall and more precision, $sim_{\alpha 2}$ and $sim_{cp_{rev}}$, will select lower frequency neighbours than the high recall measures. The measures which require both high recall and high precision, sim_{jacc}, sim_{dice}, sim_{lin} and $sim_{jacc_{mi}}$ select neighbours of a similar frequency to the target noun.

The CR framework, however, more than simulates existing measures of distributional similarity. It defines a space of distributional similarity measures that is already populated with a few named measures. By exploring the space we aimed to discover the desirable characteristics of distributional similarity measures. It may have turned out the most useful measure within this space had already been discovered. In fact, we discovered new optimal combinations of characteristics in the context of particular applications. Our goal, however, was to understand how different characteristics relate to high performance in different applications and thus explain why one measure performs better than another.

7.1.5 Chapter 5

In Chapter 5 we considered the application of distributional similarity measures in language modelling. We showed, as in previous work by Dagan et al. (1999), that it is plausible to smooth co-occurrence probability distributions using nearest neighbour techniques. The smoothed distri-

butions obtained are substantially closer to unseen empirical distributions than are the MLE distributions or distributions derived using Add1 type smoothing. Further, by not considering more neighbours once a certain condition is met, we achieve improved results and efficiency gains. The most effective stopping condition considered the amount of change to the original distribution as a function of word frequency.

We then considered the use of pseudo-disambiguation experiments to evaluate different distributional similarity measures. We saw that the experimental set-up in a pseudo-disambiguation task is very delicate and it seems virtually impossible to eliminate all possible biases from the task. However, we developed an approach intended to minimise these biases.

We used the pseudo-disambiguation task to establish properties of distributional similarity measures which might be desirable in nearest neighbour smoothing techniques. First, it appears that contexts in which a noun occurs more frequently than would be expected if nouns and contexts occurred independently are the most salient contexts for determining distributional similarity. In other words, more selective verbs do tell us more about the similarity of their arguments than less selective verbs, even if the nouns occur with the less selective verbs more frequently. Second, the difference in extent to which each word occurs in each context does not appear to be a major factor. It appears to be sufficient to know that each word has occurred in that context. Finally, a measure that can exploit the asymmetry in the concept of substitutability performs better than one that requires similarity to be symmetric. In this task, using the best performing models, neighbours that have high precision retrieval of the target noun's co-occurrences performed better than neighbours that have high recall retrieval of the target noun's co-occurrences.

This result supports our hypothesis in Chapter 1. In a language modelling application, where the task is to overcome sparse data and fill in missing information about the word, high precision neighbours are preferred. This is likely to be because they will not make many, if any, mistakes. The contexts that a high precision neighbour of word A apparently "incorrectly" retrieves, are the ones in which we expect to observe word A as more data is seen.

We also saw that performance on the pseudo-disambiguation task was lower using a semantic measure, based on the WordNet noun hierarchy, than using any of the distributional measures. This supports our hypothesis from Chapter 2 that the most semantically related words are not necessarily the best words to use in a distributional task. If the task is distributional, then (high precision) distributionally similar words should be used.

Finally in this chapter, we saw that the best-performing CRM (and parameter settings therein) as determined using the pseudo-disambiguation task (the additive MI-based CRM), was also able to achieve better coverage and higher similarity with the held-out test data than the α-skew divergence measure on the original smoothing task. From this we conclude that pseudo-disambiguation does appear to be a valid way of predicting performance on the more complex language modelling task. However, we also saw that the choice of weight function in the smoothing algorithm has a significant effect on performance for this CRM. This is an area for further research.

7.1.6 Chapter 6

In Chapter 6 we examined the ability of distributional similarity measures to predict semantic similarity. We saw that there is positive correlation between distributional similarity values and

human synonymy judgements but it is not as great as between the semantic, WordNet-based, measure and the human synonymy judgements.

We then evaluated the ability of distributional similarity measures to predict semantic similarity as defined by the WordNet-based measure. Using this task, we could establish the properties of distributional similarity measures which might be desirable in semantic applications. First, we saw that, as in the language modelling task, the best-performing distributional measures were the MI-based CRMs. This indicates that the most selective verbs are also the most salient in the context of predicting semantic similarity. Second, the difference in extent to which each word occurs in each context does not appear to be a major factor. As in the language modelling application, it appears that it is sufficient to know that each word has occurred in that context.

However, we saw that for all CRMs, and for both high and low frequency target nouns, correlation with WordNet derived neighbours is higher for low values of β than for high values of β. In other words, neighbours according to the WordNet-based measure tend to have high recall retrieval of the target noun's co-occurrences. A high value of γ leads to high performance for high frequency target nouns but poor performance for low frequency target nouns. This suggests that WordNet-derived neighbours of high frequency target nouns also have high precision retrieval of the target noun's co-occurrences whereas the WordNet-derived neighbours of low frequency target nouns do not. It also explains why measures which are closely simulated by a high value of γ in the CR framework do well for high frequency nouns but not for low frequency nouns. The best measures for this task are those that tend to select high frequency neighbours for both high frequency target nouns and low frequency target nouns.

We also saw that there appears to be a relationship between precision-recall in the CR framework and the direction of the semantic relationship between the pairs of words. In particular, high recall neighbours tend to include more hyperonyms than hyponyms and high precision neighbours tend to include more hyponyms than hyperonyms. The reason for this relationship between precision-recall and hyponymy might be that hyperonyms are more general and thus tend to occur in a greater number of distinct contexts than their hyponyms. However, there is also a strong correlation between the number of distinct contexts in which a word appears and its total frequency. Further, a very frequent word is more likely to have been seen in more of its possible contexts and therefore is likely to appear to have high recall of a related word even if there is no hyponymy relation between the word. However, we have seen in a pair of nouns where frequencies are approximately equal, the precision-recall values were able to pick out the hyponym independent of the relative frequencies. This is an area for further study.

Finally, we attempted to use a semantic application-based task to test the measures' ability to predict semantic similarity independent of human judgements. The related words that occur are ones which have first order affinities with the target noun, by the nature of the fact that they occur in the same document. However, we have seen that, for the set of words considered, the semantic similarity measure was unable to produce any increase in performance over that obtained by just considering the word itself. This suggests that second order affinities are not useful in predicting first order affinities. Consequently, it is difficult to draw conclusions from this task about the distributional similarity measures considered.

7.2 Major Contributions

Our main contribution is the development of a framework based on the concept of lexical substitutability. We have cast the problem of measuring distributional similarity as one of *co-occurrence retrieval* (CR) for which we can measure precision and recall by analogy with the way they are measured in document retrieval.

The CR framework has allowed us to systematically explore various characteristics of distributional similarity measures.

First, we have explored the merits of symmetry and asymmetry in a similarity measure by varying the relative importance attached to precision and recall. We have seen that as the distribution of word B moves away from being identical to that of word A, it can be *similar* in one of two ways: it can have occurred in generally more distinct contexts than word A; or, it can have occurred in generally less distinct contexts than word A. In the first case, we call word B a high recall neighbour of word A; and, in the second case, we call word B a high precision neighbour of word A. Through our experimental work in Chapters 5 and 6, we have shown that the kind of neighbour preferred appears to depend on the application in hand. High precision neighbours were more useful in the language modelling task of pseudo-disambiguation and high recall neighbours were more highly correlated with WordNet-derived neighbour sets. Accordingly, it appears to be important to consider whether the task is distributional or semantic. In Chapter 2, we presented a survey of potential applications and classified them according to whether we consider them to be primarily distributional or semantic. We believe that future work using distributional similarity measures in these areas will further support our hypotheses.

Second, we have explored the way in which frequency information is utilised using different co-occurrence retrieval models (CRMS). Using type-based, token-based and MI-based CRMs, we have investigated the relative importance of different co-occurrence types. The high performance of the MI-based models serves to confirm our hypothesis that highly selective verbs are more important in the calculation of distributional similarity. Using additive and difference-weighted versions of each model, we have investigated whether the difference in extent to which each noun has each feature is important. We have shown that considering this difference does not seem to be a major factor and does not lead to significantly improved results when using the best-performing MI-based CRMs.

We have also studied the tendencies of measures to select high and low frequency nouns as neighbours. We have seen that high recall measures tend to select high(er) frequency nouns as neighbours, high precision measures tend to select low(er) frequency nouns as neighbours and requiring both high recall and high precision leads to neighbours of a similar frequency to the target noun. Knowledge of these tendencies can be very important in particular applications. We are less likely to observe a single high precision neighbour than a single high recall neighbour and therefore, in general, we will need to consider a larger neighbourhood size when using high precision neighbours. We can also use these tendencies to make predictions about what will happen when we consider a greater number of words as potential neighbours. High recall neighbours will still tend to be the most frequent nouns and high precision neighbours will still tend to be the least frequent nouns. As a result, approximate neighbour sets can be computed more efficiently i.e., without considering all possible nouns as potential neighbours.

Another important contribution of this work on co-occurrence retrieval is a better understanding of existing distributional similarity measures. By comparing existing measures with the CR framework, we can analyse their CR characteristics. We have shown that six of the fifteen similarity measures considered can be directly derived or almost exactly approximated by CRMs. For example, knowing that Lin's Measure is almost equivalent to the harmonic mean of precision and recall in the additive MI-based model explains why this measure does badly on the WordNet prediction task for low frequency nouns. We have seen that recall is more important than precision in the WordNet prediction task whereas the nearest neighbours of a target noun according to Lin's Measure have both high precision and high recall. Since high precision neighbours tend to be lower frequency words, the additional requirement of high precision when considering low frequency nouns leads to the drop in performance.

Finally, this thesis has been an extensive study of both existing and new measures of distributional similarity and their applications. Our evaluation of measures has been performed over a set of 2000 nouns and we have shown that the performance of distributional similarity techniques for low frequency nouns is not significantly lower than for high frequency nouns. This suggests that distributional techniques might be used even when there is relatively little data available. In the distributional domain, this means that we can use probability estimation techniques for rare words with greater confidence. In the semantic domain, we might be able to use distributional techniques to extend existing semantic resources to cover rare or new words or automatically generate domain, genre or dialect specific resources.

7.3 Major Directions for Further Work

There are a number of major directions in which this work can be extended.

First, the set of CRMs that we have defined is not exhaustive. There are a number of other association functions, which could be used to define a CRM. For example, in Chapter 3, we discussed the potential of weighted mutual information (WMI). In other research, it has been shown that using WMI outperforms MI for low frequency words. Although, we have seen high performance for the MI-based CRMs for low frequency words, it could be that performance using WMI would be even higher.

Second, we would like to evaluate our CRMs in every application area discussed in Chapter 2. We see most potential for improvement in the tasks we have classified as distributional. In particular, we would like to try the prepositional phrase attachment ambiguity resolution task and the conjunction scope identification task. These tasks have been previously tackled using semantic classes to predict what is ultimately distributional information. Accordingly, we believe that it should be possible to do better using the CR framework. In particular, we hypothesise that a high precision setting in the additive MI-based CRM would lead to good results.

In the semantic domain, the potential seems the greatest for being able to distinguish hyponyms from hyperonyms. This could be useful in determining the compositionality of collocations. As noted by Baldwin et al. (2003), for many compositional collocations, the phrase is a hyponym of the head of the phrase. Accordingly, we would predict that, for compositional collocations, the phrase will have high precision retrieval of the co-occurrences of the head of the phrase. This should perform better than existing techniques (McCarthy et al., 2003) that consider distributional

similarity between the phrase and the head.

Finally, understanding the frequency selecting characteristics of distributional similarity measures may allow them to be used in a greater number of ways. For example, previously, a technique which compared the distributions of word senses to word forms would be biased towards the most frequently occurring sense. The arithmetic mean of precision and recall in the additive MI-based CRM, however, appears to be free from the frequency biases observed for other measures and settings and, thus, might be appropriate for use in this example.

References

Abney, S., Schapire, R. E., & 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 (EMNLP/VLC-1999).*

Austen, J. (1816). *Emma.* Penguin.

Baayen, H. (2001). *Word frequency distributions.* Dordrecht: Kluwer Academic Publishers.

Baldwin, T., Bannard, C., Tanaka, T., & Widdows, D. (2003). An empirical model of multiword expression decomposability. In *Proceedings of the ACL-2003 workshop on multiword expressions* (pp. 89–96). Sapporo, Japan.

Baldwin, T., & Bond, F. (2003a). Learning the countability of English nouns from corpus data. In *Proceedings of the 41st international conference of the Association for Computational Linguistics (ACL-2003)* (pp. 463–470). Sapporo, Japan.

Baldwin, T., & Bond, F. (2003b). A plethora of methods for learning English countability. In *Proceedings of the 2003 conference on empirical methods in natural language processing (EMNLP-2003)* (pp. 73–80). Sapporo, Japan.

Bannard, C., Baldwin, T., & Lascarides, A. (2003). A statistical approach to the semantics of verb-particle constructions. In *Proceedings of the ACL-2003 workshop on multiword expressions* (pp. 65–72). Sapporo, Japan.

Bensch, P. A., & Savitch, W. J. (1992). An occurrence-based model of word categorization. In *Proceedings of the 3rd meeting on mathematics of language.*

Bernard, J. R. L. (Ed.). (1990). *The Macquarie encyclopedic thesaurus.* Sydney, Australia: The Macquarie Library.

Berry-Rogghe, G. L. M. (1973). The computation of collocations and their relevance to lexical studies. In A. J. Aitken, R. W. Bailey, & N. Hamilton-Smith (Eds.), *The computer and literary studies.* Edinburgh, New York: University Press.

Blaheta, D., & Johnson, M. (2001). Unsupervised learning of multi-word verbs. In *In the proceedings of the 39th international conference of the Association for Computational Linguistics (ACL-2001)* (p. 54-60). Toulouse, France.

Bond, F. (2001). *Determiners and number in english, contrasted with japanese, as exemplified in machine translation.* Unpublished doctoral dissertation, University of Brisbane, Queensland, Australia.

Brill, E. (2000). Pattern-based disambiguation for natural language processing. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora (EMNLP/VLC-2000).*

Briscoe, E., & Carroll, J. (1995). Developing and evaluating a probabilistic lr parser of part-of-speech and punctuation labels. In *4th ACL/SIGDAT international workshop on parsing technologies* (pp. 48–58).

Briscoe, E., & Carroll, J. (2002). Robust accurate statistical annotation of general text. In *Proceedings of 3rd international conference on language resources and evaluation (LREC-2002)* (pp. 1499–1504). Las Palmas, Canary Islands.

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

Budanitsky, A. (1999). *Lexical semantic relatedness and its application in natural language processing.* Unpublished doctoral dissertation, Department of Computer Science, University of Toronto.

Budanitsky, A., & Hirst, G. (2001). Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the NAACL-01 workshop on WordNet and other lexical resources.* Pittsburgh.

Caraballo, S. (1999). Automatic construction of a hypernym-labelled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL-99)* (p. 120-126).

Carroll, J. (2003). *RASP parser accuracy.* personal communication. (updated version of table that appears in Carroll(1999))

Carroll, J., & Briscoe, E. (1996). Apportioning development effort in a probabilistic lr parsing system through evaluation. In *ACL/SIGDAT conference on empirical methods in natural language processing* (p. 92-100).

Carroll, J., Minnen, G., Canning, Y., Devlin, S., & Tait, J. (1998). Practical simplification of english text to assist aphasic readers. In *Proceedings of the AAAI-98 workshop on integrating artificial intelligence and assistive technology* (pp. 7–10). Madison, Wisconsin.

Charniak, E. (2000). A maximum entropy inspired parser. In *Proceedings of the 1st conference of the North American chapter of the Association for Computational Linguistics (NAACL-2000)* (pp. 132–137). Seattle, WA.

Chen, S., & Goodman, J. (1998). *An empirical study of smoothing techniques for language modelling* (Tech. Rep. No. TR-10-98). Cambridge, Massachusetts: Harvard University.

Chomsky, N. (1964). Syntactic structures. In *Janua linguarum.* The Hague, The Netherlands: Mouton.

Church, K. W. (2000). Empirical estimates of adaptation: The chance of two noriegas is closer to $p/2$ than p^2. In *Proceedings of the 18th internation conference on computational linguistics (COLING2000)* (pp. 173–179).

Church, K. W., Gale, W., Hanks, P., Hindle, D., & Moon, R. (1994). Lexical substitutability. In B. T. S. Atkins & A. Zampolli (Eds.), *Computational approaches to the lexicon* (pp. 153–177). Oxford University Press.

Church, K. W., & Hanks, P. (1989). Word association norms, mutual information and lexicography. In *Proceedings of the 27th annual conference of the Association for Computational Linguistics (ACL-1989)* (pp. 76–82).

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

Clark, A. (2000). Inducing syntactic categories by context distribution clustering. In *Proceedings of the 4th conference on natural language learning (CoNLL-2000)* (pp. 91–94). Lisbon.

Clark, E. (1992). Conventionality and contrast: Pragmatic principles with lexical consequences. In *A. lehrer and e.f. kittay, editors, frames, fields and contrasts: New essays in semantic and lexical organization.* Lawrence Erlbaum Associates.

Clark, S., & Weir, D. (2000). A class-based probabilistic approach to structural disambiguation. In *Proceedings of the 18th international conference on computational linguistics (COLING-2000)* (pp. 194–200). Saarbrucken, Germany.

Clark, S., & Weir, D. (2002). Class-based probability estimation using a semantic hierarchy. *Computational Linguistics, 28*(2).

Collins, M. (1999). *Head-driven statistical models for natural language processing.* Unpublished doctoral dissertation, Computer and Information Science, University of Pennsylvania.

Collins, M., & Brooks, J. (1995). Prepositional phrase attachment through a backed-off model. In *Proceedings of the third workshop on very large corpora* (pp. 27–38). Cambridge, Massachusetts.

Copestake, A. (1990). An approach to building the hierarchical element of a lexical knowledge base from a machine readable dictionary. In *Proceedings of the 1st international workshop on inheritance in natural language processing* (pp. 19–29). Tilburg.

Cover, T., & Thomas, J. (1991). *Elements of information theory.* New York: Wiley.

Cruse, D. (1986). *Lexical semantics.* Cambridge University Press.

Curran, J. R. (2002). Ensemble methods for automatic thesaurus extraction. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP-2002).* Philadelphia.

Curran, J. R., & Moens, M. (2002a). Improvements in automatic thesaurus extraction. In *ACL-SIGLEX workshop on unsupervised lexical acquisition.* Philadelphia.

Curran, J. R., & Moens, M. (2002b). Scaling context space. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL-2002).* Philadelphia.

Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey, J. W. (1992). Scatter/gather: A cluster-based approach to browsing large document collections. In *15th annual international SIGDIR* (pp. 318–329). Denmark.

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

Dagan, I., Marcus, S., & Markovitch, S. (1993). Contextual word similarity and estimation from sparse data. In *31st annual meeting of the Association for Computational Linguistics (acl-1993)* (pp. 56–63).

Deerwester, S., Dumais, G., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(16), 391–407.

Devlin, S. (1999). *Simplifying natural language text for aphasis readers.* Unpublished doctoral dissertation, University of Sunderland, UK.

Dhillon, I. S., Mallela, S., & Kumar, R. (2002). *Enhanced word clustering for hierarchical text classification* (Tech. Rep. Nos. TR–02-17). Austin: Department of Computer Sciences, University of Texas.

Donghong, J., Jun, H., & Changning, H. (1998). Learning new compositions from given ones. In *Proceedings of the international conference on computational natural language learning (CoNLL-1998)* (pp. 25–32). Somerset, New Jersey.

Dorr, B., & Jones, D. (1996). Acquisition of semantic lexicons: Using word sense disambiguation to improve precision. In *Proccedings of the siglex workshop on breadth and depth of semantic lexicons, 34th annual meeting of the Association for Computational Linguistics* (pp. 42–50). Santa Cruz, CA.

Essen, U., & Steinbiss, V. (1992). Cooccurrence smoothing for stochastic language modelling. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP-1992), 1*, 161–164.

Even-Zohar, Y., & Roth, D. (2000). A classification approach to word prediction. In *Proceedings of the 1st conference of the North American chapter of the Association for Computational Linguistics (NAACL-2000)* (pp. 124–131).

Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database.* MIT Press.

Finch, S. P., & Chater, N. J. (1992). Bootstrapping syntactic categories. In *Proceedings of the 14th annual conference of the cognitive science society of america* (pp. 820–825).

Firth, J. R. (1957). A synopsis of linguistic theory 1930-1955. In *Studies in linguistic analysis* (pp. 1–32). Oxford: Philogical Society. (reprinted in Palmer, F. (ed.) 1968 *Selected Papers of J.R. Firth*, Longman, Harlow)

Forgy, E. W. (1965). Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. In *Biometric society meetings.* Riverside, CA. (Abstract in *Biometrics* **21**, No. 3, 768)

Forsyte, G., Malcolm, M., & Moler, C. (1977). *Computer methods for mathematical computations.* Englewood Cliffs, New Jersey: Prentice Hall.

Frakes, W., & Baeza-Yates, R. (Eds.). (1992). *Information retrieval, data structures and algorithms.* Prentice Hall.

Friedman, N., & Singer, Y. (1999). Efficient bayesian parameter estimation in large discrete domains. *Advances in Neural Information Processing Systems 11*.

Fung, P., & McKeown, K. (1997). A technical word- and term- translation aid using noisy parallel corpora across language groups. *Machine Translation, 1–2*, 53–87.

Gale, W., Church, K. W., & Yarowsky, D. (1992). Work on statistical methods for word sense disambiguation. In *Working notes of the AAAI symposium on probabilistic approaches to natural language* (pp. 54–60). Menlo Park, CA: AAAI Press.

Gale, W., & Sampson, G. (1996). *Good-Turing estimation without tears.* COGS Research Paper.

Gibbons, J. D. (1993). Nonparametric measures of association. In *Quantitative applications in the social sciences.* Newberry Park, CA: Sage.

Golding, A. R. (1995). A bayesian hybrid method for context-sensitive spelling correction. In *Proceedings of the 3rd workshop on very large corpora* (pp. 39–53). Boston, MA.

Golding, A. R., & Roth, D. (1999). A winnow-based approach to context-sensitive spelling correction. *Machine Learning, 34*(1–3), 182–190.

Goodman, N. (1952). On likeness of meaning. In *L. linsky, editor, semantics and philosophy of language.* Urbana, Illinois: University of Illinois Press.

Grefenstette, G. (1994a). Corpus-derived first-, second- and third-order word affinities. In *Proceedings of Euralex* (p. 279-290). Amsterdam.

Grefenstette, G. (1994b). *Explorations in automatic thesaurus discovery.* Boston, USA: Kluwer Academic Publishers.

Grice, H. P. (1957). Meaning. *Philosophical Review*. (Reprinted in Strawson (1971:39–48, Philosophical Logic, Oxford University Press))

Grishman, R., & Sterling, J. (1993). Smoothing of automatically generated selectional constraints. In *Human language technology: Proceedings of the ARPA workshop* (pp. 254–259). San Francisco: Morgan Kaufmann.

Grishman, R., & Sterling, J. (1994). Generalizing automatically generated selectional patterns. In *Proceedings of the 15th international conference on computational linguistics (COLING-94)* (pp. 742–747). Kyoto, Japan.

Halliday, M. (1966). Lexis as a linguistic level. In C. Bazell, J. Catford, M. Halliday, & R. Robins (Eds.), *In memory of J. R. Firth.* London: Longman.

Harris, Z. S. (1968). *Mathematical structures of language.* New York: Wiley.

Haspelmath, M. (2002). *Understanding morphology.* Arnold Publishers.

Hatzivassiloglou, V., & McKeown, K. (1993). Towards the automatic identification of adjective scales: clustering of adjectives according to meaning. In *Proceedings of the 31st annual meeting of the Association for Computational Linguistics (ACL-1993)* (pp. 172–182).

Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th international conference on computational linguistics (COLING-92)* (pp. 539–545). Nantes, France.

Hindle, D. (1983). *User manual for fidditch.* (Naval Research Laboratory Technical Memorandum # 7590-142)

Hindle, D. (1990). Noun classification from predicate-argument structures. In *Proceedings of the 28th annual meeting of the Association for Computational Linguistics (ACL-1990)* (p. 268-275).

Hindle, D., & Rooth, M. (1991). Structural ambiguity and lexical relations. In *Proceedings of the 29th annual meeting of the Association for Computational Linguistics (ACL-1991)* (pp. 229–236). Berkeley, CA.

Hindle, D., & Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational Linguistics*, *19*(1), 103–120.

Hirschman, L., Grishman, R., & Sager, N. (1975). Grammatically-based automatic word class formation. *Information Processing and Management*, *11*, 39–57.

Hirst, G., & St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In *Fellbaum (1998)* (pp. 305–332). MIT Press.

Hofmann, T., & Puzicha, J. (1998). Mixture models for cooccurrence data. In *Proceedings of the international conference on pattern recognition.* (Longer version available as MIT A.I. Memo No.1625)

Jarmasz, M., & Szpakowicz, S. (2003). *Roget's thesaurus and semantic similarity* (Tech. Rep. No. TR-2003-01). University of Ottawa.

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the international conference on research in computational linguistics (ROCLING X).* Taiwan.

Karov, Y., & Edelman, S. (1998). Similarity-based word sense disambiguation. *Computational Linguistics*, *24*(1), 41–59.

Katz, S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-35(3)*, 400-401.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis.* John Wiley and Sons.

Kay, M., & Röscheisen, M. (1993). Text-translation alignment. *Computational Linguistics*, *19*(1), 121–142.

Keil, F. (1979). *Semantic and conceptual development: An ontological perspective.* Cambridge, Massachusetts: Harvard University Press.

Keil, F. (1983). On the emergence of semantic and conceptual distinctions. *Experimental Psychology*, *112*, 357–385.

Kilgarriff, A. (1992). *Polysemy.* Unpublished doctoral dissertation, School of Cognitive and Computing Sciences (COGS), University of Sussex.

Kilgarriff, A. (1996). Which words are particularly characteristic of a text? a survey of statistical approaches. In *Language engineering for document analysis and recognition* (pp. 33–40).

Kilgarriff, A., & Tugwell, D. (2001a). WASP-bench: an MT lexicographers' workstation supporting state-of-the-art lexical disambiguation. In *Proceedings of the machine translation (MT) summit vii.* Santiago de Compostela.

Kilgarriff, A., & Tugwell, D. (2001b). WORD SKETCH: Extraction and display of significant collocations for lexicography. In *ACL workshop on COLLOCATION: Computation extraction, analysis and exploitation.* Toulouse.

Kilgarriff, A., & Yallop, C. (2000). What's in a thesaurus. In *Second conference on language resources and evaluation (LREC-00)* (pp. 1371–1379). Athens.

Korhonen, A., Krymolowski, Y., & Marx, Z. (2003). Clustering polysemic subcategorization frame distributions semantically. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics (ACL-2003)* (pp. 64–71). Sapporo, Japan.

Kullback, S., & Leibler, R. (1951). On information and sufficiency. *Ann. Math. Stat.*, *22*, 79–86.

Kurohashi, S., & Nagao, M. (1992). Dynamic progamming method for analyzing conjunctive structures in japanese. In *Proceedings of the international conference on computational linguistics (COLING-1992).* Nantes, France.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by.* Chicago, Illinois: Chicago University Press.

Lapata, M., & Brew, C. (1999). Using subcategorization to resolve verb class ambiguity. In *Empirical methods in natural language processing.*

Lapata, M., Keller, F., & McDonald, S. (2001). Evaluating smoothing algorithms against plausibility judgements. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics (ACL-2001)* (pp. 346–353). Toulouse.

Lapata, M., & Lascarides, A. (2003). A probabilistic account of logical metonymy. *Computational Linguistics, 29*(2), 263–317.

Lapata, M., McDonald, S., & Keller, F. (1999). Determinants of adjective-noun plausibility. In *Proceedings of the 9th meeting of the European chapter of the Association for Computational Linguistics (EACL-1999)* (pp. 30–36).

Laplace, P. S. (1995). *Philosophical essay on probabilities.* Springer-Verlag.

Lauer, M. (1996). *Designing statistical language learners: Experiments on noun compounds.* Unpublished doctoral dissertation, Macquarie University.

Leacock, C., & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. In C. Fellbaum (Ed.), *WordNet: an electronic lexical database.* MIT Press.

Lee, L. (1999). Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL-1999)* (pp. 23–32).

Lee, L. (2001). On the effectiveness of the skew divergence for statistical language analysis. *Artificial Intelligence and Statistics*, 65–72.

Lee, L., & Pereira, F. (1999). Distributional similarity: Clustering vs. nearest neighbours. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL-99)* (pp. 33–40).

Lee, L. J. (1997). *Similarity-based approaches to natural language processing.* Unpublished doctoral dissertation, Harvard University, Cambridge, Massachusetts.

Levin, B. (1993). *English verb classes and alternations: A preliminary investigation.* Chicago University Press.

Li, H. (2002). Word clustering and disambiguation based on co-occurrence data. *Natural Language Engineering, 8*(1), 25–42.

Li, H., & Abe, N. (1998). Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics, 24*(2), 217–244.

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of ACL/EACL-97* (p. 64-71).

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *Proceedings of the 36th annual meeting of the Association for Computational Linguistics and the 17th international conference on computational linguistics (COLING-ACL '98)* (pp. 768–774). Montreal.

Lin, D. (1998b). Extracting collocations from text corpora. In *First workshop on computational terminology.* Montreal, Canada.

Lin, D. (1998c). An information-theoretic definition of similarity. In *Proceedings of international conference on machine learning.* Madison, Wisconsin.

Lin, D. (1999). Automatic identification of non-compositional phrases. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL-1999)* (pp. 317–324). College Park, USA.

Lin, D., & Pantel, P. (2001). Induction of semantic classes from natural language text. In *Proceedings of ACM SIGKDD conference on knowledge discovery and data mining* (pp. 317–322).

Lin, D., Zhao, S., Qin, L., & Zhou, M. (2003). Identifying synonyms among distributionally similar words. In *Proceedings of IJCAI-03* (pp. 1492–1493).

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory, 37*(1), 145–151.

Lyons, J. (1977). *Semantics.* Cambridge: Cambridge University Press.

Mangu, L., & Brill, E. (1997). Automatic rule acquisition for spelling correction. In *Proceedings of the international conference on machine learning (ICML-1997).*

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing.* Cambridge, Massachusets: MIT Press.

Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: the penn treebank. *Computational Linguistics, 19*.

Markert, K., & Nissim, M. (2002). Metonymy as a classification task. In *Proceedings of the conference on empirical methods and natural language processing (EMNLP-2002)* (pp. 204–213).

Maynard, D., & Ananiadou, S. (1998). Term sense disambiguation using a domain-specific thesaurus. In *Proceedings of first international conference on language resources and evaluation (LREC-1998).*

McCarthy, D. (2000). Using semantic preferences to identify verbal participation in role switching alternations. In *Proceedings of the first meeting of the North American chapter of the Association for Computational Linguistics (NAACL-2000).* Seattle, WA.

McCarthy, D., Keller, B., & Carroll, J. (2003). Detecting a continuum of compositionality in phrasal verbs. In *Proceedings of the ACL-2003 workshop on multiword expressions* (pp. 73–80). Sapporo, Japan.

McCarthy, D., Villavicencio, A., Korhonen, A., & Bond, F. (Eds.). (2003). *Proceedings of the ACL-2003 workshop on multiword expressions: Analysis, acquisition and treatment.* Sapporo, Japan.

Meyer, J., & Dale, R. (2002). Using the WordNet hierarchy for associative anaphora resolution. In *Proceedings of SemaNet02.*

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). Introduction to WordNet: an on-line lexical database. *Journal of Lexicography, 3*(4), 235–244. (Revised 1993, available from Princeton University)

Miller, G., Chodorow, M., Landes, S., Leacock, C., & Thomas, R. (1994). Using a semantic concordance for sense identification. In *Proceedings of the ARPA human language technology workshop.*

Miller, G. A., & Charles, W. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes, 6*(1), 1–28.

Moldovan, D., Harabagiu, S., Peters, W., Guthrie, L., & Wilks, Y. (Eds.). (2001). *WordNet and other lexical resources: Applications, extensions and customizations.* Carnegie Mellon University, Pittsburgh. (NAACL 2001 Workshop)

Morris, J., & Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics, 18*(1), 21–45.

Nissim, M., & Markert, K. (2003). Syntactic features for supervised metonymy resolution. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics (ACL-2003)* (pp. 56–63). Sapporo, Japan.

Padó, S., & Lapata, M. (2003). Constructing semantic space models from parsed corpus. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics (ACL-2003)* (pp. 128–135). Sapporo, Japan.

Pantel, P., & Lin, D. (2000a). An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of the 38th annual meeting of the Association for Computational Linguistics (ACL-2000).*

Pantel, P., & Lin, D. (2000b). Word-for-word glossing of contextually similar words. In *Proceedings of the conference on applied natural language processing / 1st meeting of the north ammerican chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)* (pp. 78–85).

Pearce, D. (2001). Synonymy in collocation extraction. In *Proceedings of the NAACL workshop on WordNet and other lexical resources: Applications, extensions and customizations.* Carnegie Mellon University, Pittsburgh.

Pearce, D. (2002). A comparative evaluation of collocation extraction techniques. In *Proceedings of the third international conference on language resources and evaluation (LREC-2002).*

Pearsall, J., & Hanks, P. (Eds.). (1998). *The new oxford dictionary of english.* Oxford University Press. (machine-readable version)

Pedersen, T. (1996). Fishing for exactness. In *Proceedings of conference on south-central SAS users group.* Texas.

Pereira, F., Tishby, N., & Lee, L. (1993). Distributional clustering of similar words. In *Proceedings of the 31st annual meeting of the Association for Computational Linguistics (ACL-1993)* (p. 183-190). Columbus, Ohio.

Preiss, J. (2000). *Word sense disambiguation using CIDE+.* Unpublished master's thesis, Engineering Department, Cambridge University.

Pustejovsky, J. (1995). *The generative lexicon.* The MIT Press.

Quine, W. (1951). Two dogmas of empiricism. *Philosophical Review, 60,* 20–43.

Quine, W. V. O. (1960). *Word and object.* Cambridge, Massachusets: MIT Press.

Quinlan, P. (1992). *The oxford psycholinguistic database.* Oxford University Press.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics, 19*(1), 17–30.

Rao, C. R. (1983). Diversity: Its measurement, decomposition, apportionment and analysis. *Sankyha: The Indian Journal of Statistics, 44(A),* 1–22.

Ratnaparkhi, A., Reynar, J., & Roukos, S. (1994). A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA workshop on human language technology.*

Ravichandran, K., & Rao, K. C. S. (2001). A new approach to fuzzy part-family formation in cellular manufacturing systems. *Advanced Manufacturing Technology*, *18*, 591–597.

Resnik, P. (1993). *Selection and information: A class-based approach to lexical relationships.* Unpublished doctoral dissertation, University of Pennsylvania.

Resnik, P. (1995). Using information content to evaluate semantic similarity. In *Proceedings of the 14th international joint conference on artificial intelligence* (pp. 448–453). Montreal.

Resnik, P. (1998). WordNet and class-based probabilities. In C. Fellbaum (Ed.), *WordNet: an electronic database.* MIT Press.

Roget, P. (1911). *Thesaurus of English words and phrases.* London, UK: Longmans, Green and Co.

Rooth, M., Riezler, S., Prescher, D., Carroll, G., & 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 (ACL-1999)* (pp. 104–111).

Rosenfeld, R. (2000). Two decades of statistical language modeling: where do we go from here? In *Proceedings of the IEEE* (Vol. 88(8)).

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, *8*(10), 627–633.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., & Flickinger, D. (2002). Multiword expressions: a pain in the neck for NLP. In *Proceedings of the 3rd international conference on intelligent text processing and computational linguistics (CICLing-2002)* (pp. 1–15). Mexico City.

Salton, G., & McGill, M. (1983). *Introduction to modern information retrieval.* McGraw-Hill.

Salton, G., Wang, A., & Yang, C. (1975). A vector-space model for information retrieval. *Journal of the American Society for Information Science*, *18*, 613–620.

Schroeder, M., & Noy, P. (2001). Multi-agent visualisation based on multivariate data. In *Proceedings of the 5th international conference on autonomous agents.*

Schulte im Walde, S. (2000). Clustering verbs semantically according to their syntactic behaviour. In *Proceedings of the 18th international conference on computational linguistics (COLING-00)* (pp. 747–753).

Schulte Im Walde, S., & Brew, C. (2002). Inducing german semantic verb classes from purely syntactic subcategorization information. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL-2002).*

Schütze, H. (1992a). Context space. In R. Goldman, P. Norvig, E. Charniak, & B. Gale (Eds.), *Working notes of the AAAI fall symposium on probabilistic approaches to natural language* (pp. 113–120). Menlo Park, CA: AAAI Press.

Schütze, H. (1992b). Dimensions of meaning. In *Proceedings of the conference on supercomputing* (pp. 787–796). Minneapolis, MN.

Schütze, H. (1993). Part-of-speech induction from scratch. In *Proceedings of the 31st annual meeting of the Association for Computational Linguistics (ACL-1993)* (pp. 251–258).

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, *24*(1), 97–124.

Shimohata, S., Sugio, T., & Nagata, J. (1997). Retrieving collocations by co-occurrences and word order constraints. In *Proceedings of the 35th meeting of the Association for Computational Linguistics (ACL-1997)* (pp. 476–481). Madrid, Spain.

Silber, H., & McCoy, K. (2002). Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, *28*(4).

Smadja, F., McKeown, K. R., & Hatzivassiloglou, V. (1996). Translating collocations for bilingual lexicons: a statistical approach. *Computational Linguistics*, *22*(1), 1–38.

Sparck-Jones, K. (1964). *Synonymy and semantic classification.* Unpublished doctoral dissertation, University of Cambridge.

Stetina, J., & Nagao, M. (1997). Corpus based PP attachment ambiguity resolution with a semantic dictionary. In *Proceedings of the fifth workshop on very large corpora* (pp. 66–80). Beijing and Hong Kong.

Stevenson, S., & Merlo, P. (1999). Automatic verb classification using distributions of grammatical features. In *Proceedings of the European chapter of the Association for Computational Linguistics (EACL-1999).* Bergen, Norway.

Stevenson, S., & Merlo, P. (2000). Automatic lexical acquisition based on statistical distributions. In *Proceedings of the 18th international conference on computational linguistics (COLING-2000).* Saarbrucken, Germany.

Strzalkowski, T., & Carballo, J. P. (1993). Recent developments in natural language text retrieval. In *Proceedings of the text retrieval conference (TREC) '93.*

Sugawara, K., Nishimura, K., Toshioka, K., Okachi, M., & Kaneko, T. (1985). Isolated word recognition using hidden markov models. In *Proceedings of the international conference on acoustics, speech and signal processing (icassp-1985)* (p. 1-4). Tampa, Florida.

Ullmann, S. (1962). *Semantics.* Oxford: Basil Blackwell.

van Rijsbergen, C. (1979). *Information retrieval* (second ed.). Butterworths.

von Leibniz, G. W. (1704). *Table de définitions.* in L. Couturat ed. 1904, Opuscules et fragments indédits de Leibniz, Paris.

Vossen, P., & Copestake, A. (1993). Untangling definition structure into knowledge representation. In E. B. et al. (Ed.), *Inheritance, defaults and the lexicon.* Cambridge University Press.

Ward, G. (1996). *Moby thesaurus.* Moby Project.

Weeds, J. (2000). *Word sense disambiguation using CIDE+*. Unpublished master's thesis, Engineering Department, Cambridge University.

Weeds, J. (2002). The reliablity of a similarity measure. In *Proceedings of the 5th uk special interest group for computational linguistics (CLUK5).* Leeds.

Weeds, J. (2003). Smoothing using nearest neighbours. In *Proceedings of the 6th uk special interest group for computational linguistics (CLUK6).* Edinburgh.

Weeds, J., & Weir, D. (2003a). Finding and evaluating sets of nearest neighbours. In *Proceedings of the 2nd international conference on corpus linguistics.* Lancaster, UK.

Weeds, J., & Weir, D. (2003b). A general framework for distributional similarity. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP-2003).* Sapporo, Japan.

Wiebe, J. (2000). Learning subjective adjectives from corpora. In *AAAI '00.*

Wu, H., & Zhou, M. (2003). Synonymous collocation extraction using translation information. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics (ACL-2003)* (pp. 120–127). Sapporo, Japan.

Yarowsky, D. (1993). One sense per collocation. In *Proceedings of the ARPA human language technology workshop.* Princeton.

Yarowsky, D. (1994). A comparison of corpus-based techniques for restoring accents in Spanish and French text. In *Proceedings of the 2nd workshop on very large corpora.* Kyoto, Japan.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivalling supervised methods. In *Proceedings of the 33rd annual meeting of the Association for Computational Linguistics (ACL-95)* (pp. 189–196).

Zipf, G. (1949). *Human behaviour and the principle of least effort.* Addison-Wesley.

Index

absolute value distance, 48
additive co-occurrence retrieval models, 76
additive MI-based CRM, 79
additive token-based CRM, 78
additive type-based CRM, 77
α-skew divergence measure, 127
α-skew divergence measure, 55, 73, 100, 117
anaphor resolution, 31
antonymy, 18, 31
applications, 22–36
arithmetic mean (of precision and recall), 83
asymmetry in similarity, 22
automatic thesaurus generation, 20, 30

bias towards high frequency neighbours ($biashigh_{sim}$), 70
boosting, 25

city-block distance, 48
clustering, 35
    agglomerative hierarchical, 35
    hard, 35
    k-means, 35
    nearest-neighbour, 35
    soft, 35
clustering vs. *k* nearest neighbours, 35
co-occurrence retrieval, 75
    high precision, 91
    high recall, 153
    htigh precision, 139
co-occurrence retrieval high recall, 91
co-occurrence retrieval models (CRMs), 76–82
co-occurrence type, 46
collocation extraction, 32
compositional collocations, 32
compound noun interpretation, 34
confusion probability, 57, 102
conjunction scope identification, 26
correlation coefficients, 50
cosine measure, 49, 95
countability, 29

decision lists, 25
Dice coefficient, 53, 97
difference-weighted CRMs, 79
difference-weighted MI-based CRM, 82
difference-weighted token-based CRM, 81
difference-weighted type-based CRM, 80
dissimilarity measures, 46
distance measures, 46
distance threshold, 127
distance-weighted averaging, 36, 114
    stopping conditions, 116
distribution width, 70
distributional hypothesis, 19
divergence measures, 46

Euclidean distance, 48
evaluation
    against WordNet, 43, 151–162
    application-based, 40
    gold standard, 36
    human plausibility judgements, 38
    pseudo-disambiguation, 41
    unseen data, 39
evaluation of distributional similarity measures, 36

F-score, 83

geometric measures, 47
grammatical dependency relation data, 18

harmonic mean (of precision and recall), 83
Hindle's measure, 58, 106
human synonymy judgements, 142
hyponymy, 18, 31, 148, 155

information retrieval, 31

Jaccard's coefficient, 53, 97, 109

Jensen-Shannon divergence measure, 55, 99

k nearest neighbours, 123
Kendall's τ coefficient, 52, 96
Kullback-Leibler divergence measure, 54

L_1 Norm, 48, 93, 127
L_2 Norm, 48, 95
language modelling, 24
latent semantic indexing (LSI), 18
lexical chaining, 34
lexical cohesion, 163
lexical substitutability, 22
Lin's MI-based measure, 59, 106, 127
Lin's similarity theorem, 59
Lin's WordNet based similarity measure, 127
looser thesaurus, 18

Macquarie Thesaurus, 21
Manhattan Distance, 48
matching coefficient, 54
Maximum entropy, 25
maximum likelihood estimation, 118
metonymy resolution, 29
Minkowski Distance, 47
Moby Thesaurus, 21
multiword expression analysis, 32
mutual information (MI), 57

nearest neighbour, 46
nearest neighbour smoothing, 112–122
neighbour set comparison technique, 65
non-composition collocations, 32

Pearson's product-moment correlation coefficient, 50
potential neighbour, 46
precision, 75
prepositional phrase attachment ambiguity resolution, 25
pseudo-disambiguation, 41, 122–140
  nearest neighbour voting, 122, 124–125
    frequency-weighted voting, 123
    one vote per neighbour, 123
  parameter optimisation, 127
  test set construction, 122, 125–127

RASP (Robust Accurate Statistical Parser), 60
recall, 75
relative entropy, 54, 117
Roget's Thesaurus, 21
Russell and Rao's coefficient, 54

second order affinity, 17
similarity threshold, 123, 127
singular value decomposition (SVD), 49
smoothing, 24
  add1, 118
  addX for N, 118
  algorithm for, 114
  Bayesian, 118
  evaluation, 117
  nearest neighbour, 112–122
Spearman's rank correlation coefficient, 52
spelling correction, 34, 162–169
symmetry in similarity, 21
synonymy, 17, 31

Tanimoto coefficient, 53
target noun, 46
taxi-cab distance, 48
text cohesion, 34
text simplification, 33
tighter thesaurus, 18
topic identification, 34

weighted mutual information (WMI), 59
word sense disambiguation, 27
word sense separation, 28
WordNet, 18
WordNet based similarity measures, 143
WordNet prediction task, 43

Appendix A

Sets of Nouns Used in Experiments

This appendix lists the high and low frequency sets of nouns. The figures in brackets are the associated frequencies of the nouns in the BNC data used. Each list is in order of frequency.

A.1 High Frequency Nouns

way (20871), time (18649), place (15181), hand (13803), problem (12607), head (12364), thing (12347), people (12117), part (11403), number (11251), work (11151), effect (10512), life (9826), eye (9626), man (9422), child (9337), money (9118), idea (8857), information (8279), system (8201), job (8178), power (8130), interest (8054), word (7903), year (7893), question (7662), service (7560), name (7486), day (7317), need (7164), door (7154), role (7149), point (7139), change (6944), attention (6936), view (6837), house (6678), face (6475), support (6042), use (6017), decision (5788), position (5774), sense (5767), form (5751), chance (5665), opportunity (5503), book (5437), home (5345), cost (5285), area (5240), room (5174), woman (5168), value (5144), plan (5136), line (5134), car (5093), level (5054), case (5004), arm (4957), rate (4904), policy (4894), control (4874), action (4867), mind (4710), lot (4687), company (4590), range (4574), body (4522), group (4464), business (4456), letter (4446), advantage (4439), experience (4358), government (4346), order (4345), result (4328), evidence (4311), difference (4292), relationship (4193), course (4176), account (4078), kind (4072), programme (4041), feeling (4039), friend (4015), price (4003), issue (3992), report (3979), party (3969), look (3909), game (3899), difficulty (3896), story (3872), responsibility (3853), amount (3845), water (3823), picture (3817), record (3757), world (3730), side (3701), end (3675), step (3646), access (3635), process (3582), development (3581), detail (3573), figure (3496), right (3491), night (3473), care (3450), reason (3449), claim (3444), member (3441), pattern (3423), fact (3384), skill (3382), school (3363), voice (3352), benefit (3342), law (3328), authority (3325), meeting (3313), country (3303), family (3290), hair (3274), light (3267), choice (3259), foot (3247), condition (3234), force (3227), week (3226), activity (3218), standard (3215), situation (3183), advice (3176), food (3172), approach (3148), matter (3083), note (3081), share (3074), mother (3073), material (3060), effort (3054), office (3041), risk (3026), statement (3011), quality (2998), piece (2996), sort (2995), structure (2993), agreement (2950), knowledge (2949), answer (2948), product (2909), charge

(2899), hour (2894), state (2886), rule (2880), method (2878), loss (2849), market (2846), impact (2844), demand (2844), rise (2834), message (2829), application (2829), leg (2828), finger (2763), list (2761), thought (2744), influence (2738), profit (2734), datum (2719), term (2710), girl (2710), set (2702), possibility (2701), performance (2684), back (2666), operation (2657), image (2623), function (2623), example (2616), pressure (2599), degree (2595), help (2567), deal (2567), team (2558), type (2552), goal (2545), contract (2545), contribution (2542), ball (2527), resource (2525), response (2513), glass (2499), weight (2489), property (2484), series (2480), scheme (2479), sound (2478), page (2474), model (2459), movement (2456), ability (2441), importance (2437), task (2436), road (2429), meaning (2413), impression (2400), test (2389), study (2385), land (2384), history (2364), facility (2364), bit (2358), language (2357), table (2346), father (2333), copy (2332), person (2331), sign (2315), tax (2311), wife (2303), space (2287), duty (2287), contact (2276), rest (2262), son (2246), behaviour (2243), success (2239), horse (2238), proposal (2237), baby (2235), basis (2231), argument (2231), practice (2229), breath (2224), staff (2220), attack (2218), status (2212), relation (2202), increase (2196), title (2189), subject (2187), principle (2181), event (2177), seat (2171), nature (2171), feature (2167), fire (2164), theory (2149), attitude (2149), element (2118), couple (2105), love (2100), treatment (2098), requirement (2095), month (2093), strength (2092), war (2091), confidence (2086), option (2083), cup (2072), technique (2069), provision (2067), fund (2067), training (2066), attempt (2063), class (2046), aspect (2042), concern (2038), bill (2038), truth (2032), section (2031), understanding (2025), student (2022), ground (2021), variety (2016), balance (2009), version (1997), chapter (1997), death (1994), police (1989), news (1986), measure (1981), building (1979), key (1977), moment (1973), commitment (1972), colour (1971), career (1969), period (1954), heart (1954), clothes (1954), character (1954), solution (1948), patient (1947), card (1945), million (1935), growth (1934), damage (1932), dog (1930), energy (1928), trouble (1920), call (1919), drink (1908), daughter (1902), act (1900), lip (1899), return (1888), cause (1887), boy (1886), project (1883), progress (1883), campaign (1880), parent (1879), machine (1878), goods (1877), fear (1868), environment (1867), paper (1865), appeal (1854), music (1851), intention (1851), payment (1849), visit (1843), window (1840), unit (1835), sentence (1828), opinion (1827), air (1823), style (1809), path (1806), field (1802), court (1794), club (1774), move (1770), bag (1770), implication (1767), source (1761), arrangement (1759), talk (1758), procedure (1755), sale (1744), notice (1743), mouth (1742), plant (1741), shape (1738), wall (1730), industry (1730), strategy (1729), memory (1727), hope (1723), site (1722), future (1722), mistake (1721), hold (1708), concept (1703), design (1701), education (1697), lead (1695), player (1694), reputation (1685), bed (1674), stage (1670), research (1662), centre (1662), shoulder (1656), offer (1656), doctor (1647), film (1639), minute (1623), item (1612), smile (1602), analysis (1601), church (1587), pain (1586), post (1583), appearance (1578), fish (1576), society (1575), connection (1573), worker (1571), match (1570), animal (1569), income (1567), expression (1566), scene (1562), document (1556), equipment (1552), manager (1550), box (1546), size (1544), purpose (1540), potential (1538), route (1535), comment (1524), meal (1523), bottle (1519), council (1518), discussion (1517), tea (1512), collection (1512), firm (1507), threat (1501), shop (1501), tree (1500), instruction (1499), user (1491), production (1490), street (1487), objective (1483), investment (1483), election (1483), economy (1480), bank (1480), edge (1475), tooth (1467), protection (1466), egg

(1462), presence (1461), means (1460), date (1459), target (1453), teacher (1452), pair (1450), secretary (1448), consequence (1448), competition (1443), morning (1439), capacity (1438), challenge (1435), belief (1434), article (1431), technology (1430), vote (1429), significance (1422), proportion (1421), conclusion (1414), doubt (1413), husband (1411), hole (1405), gun (1404), mark (1402), shot (1401), board (1399), direction (1397), length (1393), age (1388), freedom (1386), existence (1385), sight (1377), file (1377), explanation (1377), distance (1368), customer (1363), city (1362), injury (1361), gap (1352), corner (1351), evening (1349), community (1345), coffee (1345), majority (1344), top (1335), conference (1334), affair (1328), committee (1326), army (1326), security (1317), object (1317), debt (1312), reaction (1310), garden (1310), base (1305), aid (1301), holiday (1297), stock (1290), neck (1286), brother (1285), start (1284), conversation (1284), trade (1283), show (1282), nose (1281), award (1281), audience (1281), train (1280), desire (1274), text (1269), noise (1268), plenty (1267), speech (1263), package (1262), extent (1262), town (1258), cut (1256), pleasure (1247), limit (1247), reader (1242), song (1236), reference (1236), minister (1236), peace (1235), officer (1235), sum (1225), pupil (1224), chair (1224), photograph (1222), flower (1218), run (1213), distinction (1211), priority (1209), boat (1203), defence (1200), management (1198), victory (1196), cash (1190), phone (1189), client (1189), factor (1187), track (1184), network (1183), speed (1176), links (1165), address (1165), supply (1163), mile (1161), disease (1161), department (1159), race (1158), improvement (1158), past (1155), leader (1154), cigarette (1150), appointment (1150), hundred (1146), skin (1144), population (1144), fee (1143), permission (1142), papers (1136), lesson (1133), computer (1133), exercise (1130), trip (1128), stair (1127), faith (1127), investigation (1124), stone (1123), battle (1121), suggestion (1119), characteristic (1114), station (1112), health (1111), habit (1111), drug (1111), weapon (1110), sheet (1110), independence (1110), sister (1104), bird (1104), initiative (1103), division (1099), assessment (1099), spirit (1098), blood (1096), obligation (1095), review (1093), opposition (1091), employment (1090), budget (1090), atmosphere (1090), flow (1086), association (1085), alternative (1083), visitor (1080), turn (1076), dress (1076), identity (1074), hat (1073), vision (1068), engine (1068), notion (1067), journey (1064), surface (1063), dream (1062), cell (1061), dinner (1058), floor (1053), cover (1051), loan (1050), capital (1045), approval (1042), entry (1038), ear (1038), request (1033), pace (1032), representation (1030), emphasis (1029), criticism (1029), band (1026), round (1024), flat (1024), sex (1023), suit (1022), play (1015), art (1015), assistance (1012), shoe (1010), organisation (1009), warning (1006), debate (1006), volume (1003), row (1002), taste (998), prime (996), half (996), wine (995), signal (995), framework (994), ship (993), recommendation (993), reduction (990), coat (990), program (989), break (986), knife (983), touch (982), criterion (982), tension (981), description (981), crime (981), danger (980), will (979), liability (976), season (975), burden (974), aim (973), awareness (969), passage (963), fortune (963), credit (962), reality (961), judgement (960), ticket (959), breakfast (958), expectation (955), survey (954), interview (954), complaint (954), lack (953), director (953), individual (952), legislation (951), promise (949), reading (946), vehicle (944), software (944), ring (943), link (941), score (940), newspaper (939), licence (939), code (939), lunch (936), offence (935), flight (935), trend (934), sequence (929), wave (926), consideration (926), stuff (923), foundation (922), tradition (920), commission (920), secret (917), definition (916), map (914), assumption (914), output (913), selection (912), culture (912), bar (912), grant (907), river (906),

driver (906), recognition (905), release (899), asset (899), distribution (898), quantity (897), labour (897), reform (896), component (896), concentration (895), heat (894), conflict (892), root (891), relief (888), partner (888), cheek (888), wish (887), victim (887), failure (885), employee (885), gift (884), reply (882), mass (880), fall (879), error (878), fruit (876), spot (875), insight (875), fun (875), session (874), plate (873), trial (872), birth (872), prize (871), tear (870), examination (870), sample (869), candidate (868), talent (865), search (865), outcome (865), hospital (864), crisis (864), tendency (861), painting (858), mechanism (858), drive (858), village (854), tool (854), introduction (854), troops (850), penalty (850), tone (849), inquiry (847), total (843), living (843), scope (841), bid (838), branch (837), pound (834), respect (831), efficiency (828), jacket (826), fight (823), tour (821), laugh (821), grip (815), finding (811), circumstance (809), origin (808), interpretation (808), leaf (805), sympathy (801), union (799), shirt (799), lady (792), region (790), theme (789), possession (789), video (788), courage (788), tie (787), resistance (786), content (786), accommodation (784), boot (783), display (781), crowd (781), profile (778), variation (777), treaty (776), bridge (770), comparison (768), temperature (766), marriage (766), exhibition (765), throat (763), thinking (762), contents (760), cheque (760), background (760), resolution (759), star (758), regulation (758), involvement (758), invitation (758), string (757), expenditure (757), enemy (755), barrier (753), trust (751), cat (751), institution (750), scale (746), boundary (746), agent (746), gaze (744), muscle (743), ban (742), harm (740), glance (740), telephone (739), wealth (736), certificate (734), communication (733), wing (732), qualification (732), expertise (732), aircraft (732), combination (731), accident (730), remark (729), organization (729), stake (728), device (728), block (728), tape (727), gate (727), guidance (725), load (723), emotion (723), pool (722), hotel (722), proceedings (719), front (717), transfer (716), symptom (716), pension (714), drop (714), soul (712), recovery (712), representative (711), hill (711), trick (710), television (710), judgment (710), indication (710), island (708), instrument (707), magazine (705), colleague (705), beginning (703), experiment (702), estate (702), brain (701), comfort (700), kid (697), joke (697), president (696), rock (694), oil (694), bus (694), tongue (691), cake (691), milk (690), restriction (689), deficit (689), height (688), wheel (687), owner (687), curtain (685), objection (682), module (681), guide (681), revenue (680), wind (679), safety (679), press (678), pipe (678), diet (678), decline (678), violence (677), chain (677), dimension (675), command (675), membership (674), hall (674), ambition (674), anger (673), shift (672), library (672), wood (670), motion (670), supplies (668), pride (668), knee (667), frame (667), preference (665), circle (665), plane (664), wages (663), sleep (662), imagination (662), bone (662), strike (661), debut (659), channel (659), sea (657), dispute (657), premise (656), perspective (656), reward (655), mood (655), species (653), protest (653), watch (652), compensation (651), seed (650), formation (650), focus (650), winner (645), phrase (645), agency (644), strain (643), creation (642), privilege (640), estimate (639), century (639), allowance (639), rent (638), afternoon (637), whole (634), silence (634), championship (634), bomb (634), tale (633), guest (632), farm (631), shadow (630), mixture (630), effectiveness (630), constitution (630), prospect (626), gain (626), camera (625), tank (624), incident (624), bell (624), clue (623), parliament (621), exchange (619), thousand (618), establishment (618), consent (618), observation (617), regard (616), location (616), guideline (616), gas (616), politics (614), weekend (613), university (613), meat (613), expense (613), category (613), suspicion (612), receiver (611), stick (610), public (609), eyebrow (609), enthusiasm (608),

incentive (607), deposit (607), poem (606), bath (606), anxiety (606), settlement (605), factory (605), excuse (605), enquiry (604), check (604), struggle (602), earth (602), store (601), clause (601), bond (601), cap (599), beer (599), science (596), master (596), conviction (596), blow (596), waste (595), stress (595), beauty (594), opening (591), satisfaction (589), generation (589), yard (588), football (588), weakness (587), flavour (587), prisoner (586)

A.2 Low Frequency Nouns

stain (121), signing (121), reactor (121), penetration (121), patrol (121), newcomer (121), trench (120), stab (120), reconstruction (120), progression (120), plaque (120), pepper (120), patronage (120), optimism (120), lining (120), gravity (120), continent (120), composure (120), waiter (119), terminology (119), plaster (119), paw (119), mayor (119), lesion (119), heed (119), fulfilment (119), feasibility (119), embassy (119), well (118), realm (118), pose (118), node (118), loaf (118), facet (118), cheer (118), vulnerability (117), ulcer (117), theology (117), supremacy (117), sterling (117), quarry (117), perfume (117), para (117), misgiving (117), flap (117), economics (117), dock (117), differentiation (117), clutch (117), beef (117), trauma (116), stamen (116), shortcoming (116), saving (116), readiness (116), primacy (116), obsession (116), meter (116), manufacture (116), fume (116), footpath (116), daylight (116), congestion (116), blend (116), blast (116), barn (116), token (115), tenancy (115), stripe (115), shoot (115), safeguard (115), reinforcement (115), rehearsal (115), projection (115), precision (115), paradox (115), foothold (115), contamination (115), brigade (115), bidding (115), arthritis (115), reversal (114), recruit (114), rag (114), qualm (114), passing (114), pang (114), multitude (114), leisure (114), kilometre (114), federation (114), deployment (114), aura (114), wares (113), vitamin (113), velocity (113), vegetation (113), trustee (113), tribe (113), tomb (113), tantrum (113), self-esteem (113), romance (113), refund (113), manipulation (113), icing (113), dwelling (113), calendar (113), banker (113), aeroplane (113), wolf (112), spur (112), shipment (112), shed (112), saucer (112), renewal (112), porter (112), outrage (112), mound (112), merchant (112), lobby (112), hay (112), corn (112), bomber (112), anomaly (112), treat (111), stove (111), simplicity (111), shareholding (111), recruitment (111), reconciliation (111), mutation (111), misunderstanding (111), grandchild (111), civilian (111), bitterness (111), arse (111), algorithm (111), wait (110), truce (110), thickness (110), temperament (110), slab (110), self-confidence (110), reservoir (110), redress (110), racism (110), penis (110), nutrient (110), monitoring (110), liquidity (110), classroom (110), spice (109), shotgun (109), shah (109), pointer (109), liver (109), groove (109), grid (109), flair (109), fax (109), excursion (109), equaliser (109), enzyme (109), dome (109), diversion (109), contention (109), commissioner (109), bacon (109), apron (109), suspect (108), singles (108), severity (108), marking (108), grouping (108), elite (108), dot (108), decrease (108), crumb (108), comb (108), carrot (108), binoculars (108), whisper (107), underground (107), trainee (107), surgeon (107), rim (107), respects (107), receptor (107), livelihood (107), lap (107), keeper (107), instability (107), ink (107), inadequacy (107), guardian (107), grudge (107), gradient (107), dart (107), ballot (107), arena (107), tobacco (106), thunder (106), syllabus (106), spin (106), quest (106), healing (106), happening (106), grape (106), glider (106), garlic (106), friction (106), footing (106), ethic (106), decay (106), compass (106), buzz (106), boiler (106), bloom (106), blockade (106), bark (106), amenity (106), acclaim (106), tan (105), squadron (105), sailor (105), reign

(105), overhead (105), mechanic (105), lace (105), intruder (105), insulation (105), grenade (105), fright (105), discrepancy (105), discontent (105), deadlock (105), dagger (105), abundance (105), rocket (104), resort (104), referral (104), pilgrimage (104), outsider (104), outbreak (104), obedience (104), idiot (104), goose (104), fort (104), forgiveness (104), embargo (104), diamond (104), desert (104), counsellor (104), cough (104), constituent (104), cabin (104), briefing (104), boycott (104), appliance (104), anguish (104), amusement (104), trout (103), taxpayer (103), surrender (103), shudder (103), scepticism (103), richness (103), purity (103), performer (103), overall (103), lounge (103), fridge (103), extreme (103), dissatisfaction (103), detour (103), corps (103), bladder (103), backbone (103), violin (102), torture (102), stir (102), sole (102), ounce (102), occupant (102), metabolism (102), masterpiece (102), knickers (102), interviewer (102), honey (102), gravel (102), foul (102), feeding (102), distributor (102), centenary (102), blueprint (102), beating (102), wheelchair (101), spade (101), sickness (101), shrine (101), rainbow (101), push (101), pigeon (101), petal (101), outing (101), obstruction (101), marble (101), liaison (101), gum (101), geography (101), functioning (101), fracture (101), despair (101), choir (101), wagon (100), turkey (100), torque (100), terrorist (100), talking (100), spark (100), sailing (100), remnant (100), realism (100), plank (100), physics (100), opposite (100), omission (100), linen (100), impatience (100), haven (100), fellowship (100), familiarity (100), draught (100), detection (100), classic (100), can (100), tourism (99), textbook (99), telescope (99), scholar (99), ramp (99), propensity (99), parity (99), monk (99), microcomputer (99), manifestation (99), making (99), lyric (99), lordship (99), longing (99), lavatory (99), individuality (99), giggle (99), generator (99), disappearance (99), dependency (99), clip (99), bribe (99), alignment (99), wheat (98), vowel (98), sunshine (98), rhetoric (98), refreshment (98), ration (98), pretension (98), pitfall (98), overdraft (98), migration (98), irony (98), helper (98), gloom (98), endeavour (98), dedication (98), cube (98), congratulation (98), compatibility (98), cane (98), bail (98), warranty (97), utmost (97), scorn (97), ripple (97), repeat (97), prohibition (97), planner (97), nationality (97), livestock (97), jeep (97), imagery (97), hatch (97), fountain (97), flank (97), fig (97), diarrhoea (97), demon (97), copper (97), cohesion (97), bouquet (97), archive (97), amnesty (97), well-being (96), vault (96), urine (96), terrorism (96), spectator (96), referee (96), racket (96), pinch (96), nickname (96), nephew (96), mole (96), mantle (96), kidney (96), flush (96), enhancement (96), donkey (96), dive (96), dip (96), courtyard (96), administrator (96), ad (96), unease (95), underwear (95), tug (95), template (95), spreadsheet (95), rift (95), replay (95), prototype (95), pillar (95), permit (95), librarian (95), intimacy (95), inconsistency (95), imprint (95), fin (95), ferret (95), duplication (95), drainage (95), displeasure (95), cruelty (95), unionist (94), transcription (94), tally (94), supermarket (94), stare (94), sponge (94), selling (94), seizure (94), salvation (94), resonance (94), psychologist (94), pebble (94), monarchy (94), landmark (94), em (94), dissent (94), cotton (94), commodity (94), backdrop (94), ant (94), alga (94), woodland (93), swim (93), skiing (93), shopper (93), remission (93), refinement (93), promoter (93), kindness (93), immigrant (93), folly (93), errand (93), erection (93), dam (93), cocktail (93), burglary (93), bracket (93), bra (93), bitch (93), acknowledgement (93), wig (92), vector (92), tram (92), suitability (92), sherry (92), roar (92), pest (92), pastry (92), ocean (92), mattress (92), kitten (92), insider (92), inconvenience (92), hide (92), groan (92), gamble (92), fog (92), eating (92), crusade (92), conspiracy (92), condemnation (92), chore (92), bump (92), basics (92), actress (92), virgin (91), vigour (91), twinge (91), sur-

name (91), scanner (91), reunion (91), potency (91), parasite (91), nipple (91), monastery (91), mob (91), membrane (91), mastery (91), lipstick (91), hp (91), hillside (91), gauntlet (91), fringe (91), dinghy (91), cellar (91), abandonment (91), warrior (90), voyage (90), typewriter (90), rite (90), propaganda (90), printing (90), loneliness (90), gin (90), foliage (90), flask (90), explosive (90), dough (90), confidentiality (90), communist (90), ass (90), applause (90), addiction (90), worship (89), verb (89), syringe (89), stroll (89), solvent (89), shrub (89), sensibility (89), salute (89), robbery (89), recall (89), putt (89), pea (89), pasture (89), larva (89), interruption (89), indifference (89), honesty (89), ham (89), generalization (89), constant (89), compartment (89), chick (89), boredom (89), adjective (89), vet (88), update (88), therapist (88), sparkle (88), runway (88), rivalry (88), relic (88), recipient (88), paste (88), nappy (88), inn (88), glare (88), fortress (88), due (88), diameter (88), colouring (88), bulletin (88), blush (88), apprehension (88), anchor (88), ale (88), ace (88), void (87), vigil (87), vent (87), twig (87), toxin (87), thirst (87), stump (87), presumption (87), peg (87), neutrality (87), mist (87), liquor (87), hello (87), gloss (87), frog (87), excellence (87), dentist (87), deduction (87), contingent (87), chess (87), canon (87), bandage (87), visibility (86), third (86), spike (86), slate (86), scalp (86), revolver (86), radiator (86), oppression (86), leverage (86), knuckle (86), jersey (86), jail (86), imposition (86), gardening (86), filling (86), cone (86), comprehension (86), cocaine (86), coating (86), clump (86), clout (86), click (86), caller (86), symphony (85), steward (85), spotlight (85), scratch (85), rotation (85), retailer (85), reach (85), pop (85), patron (85), paperwork (85), overtone (85), mosaic (85), moor (85), mentality (85), investigator (85), intrusion (85), glue (85), dye (85), deprivation (85), broadcasting (85), bowler (85), transcript (84), telly (84), survivor (84), spiral (84), schooling (84), rebate (84), outburst (84), orbit (84), mapping (84), imprisonment (84), freeze (84), footprint (84), enforcement (84), differential (84), crest (84), crate (84), consortium (84), complement (84), coil (84), clone (84), calcium (84), alley (84), activist (84), vaccine (83), travelling (83), transfusion (83), tractor (83), sticker (83), sophistication (83), self-control (83), rigour (83), readership (83), planting (83), plain (83), menace (83), insistence (83), gleam (83), extinction (83), embryo (83), ditch (83), cruise (83), broker (83), blue (83), bloke (83), altar (83), waterfall (82), spy (82), shake (82), overcoat (82), ordeal (82), inscription (82), inflammation (82), gland (82), dawn (82), wreath (81), ventilation (81), upheaval (81), turning (81), tuition (81), socialist (81), slipper (81), shawl (81), saddle (81), remuneration (81), persona (81), perfection (81), neglect (81), napkin (81), messenger (81), linkage (81), likeness (81), hostel (81), hollow (81), freight (81), fossil (81), flute (81), fingernail (81), elegance (81), dynamics (81), devotion (81), deterrent (81), cop (81), convenience (81), cardigan (81), burglar (81), buck (81), barrage (81), staining (80), pyramid (80), pedigree (80), packaging (80), overtime (80), overlap (80), oak (80), negligence (80), mummy (80), jargon (80), homosexuality (80), harness (80), drag (80), dismay (80), cooker (80), cock (80), clay (80), cannon (80), brace (80), bargaining (80), aroma (80), aquarium (80), antibiotic (80), trumpet (79), swig (79), supervisor (79), subtlety (79), substitution (79), strawberry (79), stigma (79), rack (79), playing (79), perimeter (79), paradigm (79), packing (79), necklace (79), nationalism (79), moan (79), memo (79), local (79), litigation (79), imitation (79), haircut (79), glamour (79), foil (79), fireplace (79), elevation (79), detachment (79), capture (79), canopy (79), adherence (79), stirring (78), semblance (78), respite (78), ratification (78), pronunciation (78), pneumonia (78), onslaught (78), mockery (78), manpower (78), mammal (78), loophole (78), lighter (78), legion

(78), instalment (78), inkling (78), hound (78), fuse (78), fuck (78), fragrance (78), formality (78), float (78), fingerprint (78), exhibit (78), ecstasy (78), distaste (78), comrade (78), compost (78), compiler (78), bonfire (78), slump (77), rot (77), regression (77), prostitute (77), meadow (77), mansion (77), mainframe (77), irregularity (77), ion (77), hormone (77), harassment (77), gulp (77), frown (77), fisherman (77), cutter (77), cork (77), collision (77), captive (77), bible (77), berry (77), baton (77), witch (76), tempo (76), showing (76), schema (76), roller (76), realization (76), raincoat (76), pyjamas (76), prose (76), plague (76), piss (76), pawn (76), ore (76), noun (76), mischief (76), mining (76), majesty (76), jealousy (76), instant (76), housekeeper (76), hem (76), haemorrhage (76), globe (76), gender (76), gauge (76), gardener (76), dumping (76), downfall (76), digit (76), depot (76), demolition (76), cuff (76), compulsion (76), coastline (76), butt (76), brewery (76), beak (76), artefact (76), arc (76), airfield (76), abbey (76), wreck (75), wear (75), vine (75), valuable (75), tunic (75), terrain (75), spanner (75), softness (75), sediment (75), screening (75), provider (75), programming (75), principal (75), photography (75), pal (75), km (75), intuition (75), intent (75), generosity (75), flu (75), fingertip (75), dump (75), diesel (75), cult (75), complexion (75), commentator (75), catering (75), cancellation (75), bonnet (75), birdie (75), annoyance (75), walker (74), vice (74), uniqueness (74), tart (74), tackle (74), stadium (74), rubber (74), razor (74), questioning (74), pricing (74), pram (74), overture (74), ornament (74), magnet (74), liberal (74), leukaemia (74), innings (74), hydrogen (74), garrison (74), fungus (74), fatigue (74), detention (74), demonstrator (74), daddy (74), communion (74), colon (74), colitis (74), carbon (74), cache (74), breeze (74), amp (74), vocation (73), uniformity (73), superior (73), sitting (73), senator (73), reshuffle (73), professionalism (73), notation (73), neighbourhood (73), lodge (73), loading (73), lab (73), ingenuity (73), horde (73), hangover (73), handwriting (73), forearm (73), fair (73), ethos (73), escort (73), donor (73), distraction (73), disparity (73), disciple (73), communism (73), climber (73), authenticity (73), salon (72), resilience (72), reprieve (72), prerogative (72), predicament (72), pellet (72), peach (72), interpreter (72), inmate (72), hare (72), grounding (72), galaxy (72), fill (72), embankment (72), dissertation (72), disgust (72), bum (72), borough (72), bombing (72), bluff (72), binding (72), backlash (72), artwork (72), anorak (72)