



## **Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin**

### **Copyright statement**

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

### **Liability statement**

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

### **Access Agreement**

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

**On Coordination Disambiguation in a Generative  
Parsing Model, with Memory-Based Techniques for  
Parameter Estimation**

by

**Deirdre Hogan**

**Dissertation**

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

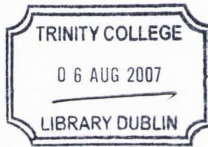
for the Degree of

**Doctor of Philosophy**

**University of Dublin, Trinity College**



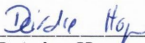
May 2007



*THESIS*  
*8182*

## Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

  
Deirdre Hogan

---

May 20, 2007

## Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Deirdre Hogan  
Deirdre Hogan

May 20, 2007

# Acknowledgments

I would like first to thank my supervisors, Pádraig Cunningham and Saturnino Luz, for their feedback and good advice over the years and for always pointing me in the right direction when things went awry.

Much thanks also to Jennifer Foster for her encouragement, for her help structuring this thesis and for the valuable and in-depth feedback she gave on the individual thesis chapters.

Thanks also to Carl Vogel for his support and encouragement, for always giving me the sense he had confidence in my ability and for organising the computational linguistics reading group. My gratitude also extends to those attending the reading group sessions especially for their comments and thoughts on coordination disambiguation.

I would also like to thank Alexey Tsymbal and Joachim Wagner for their useful feedback on parts of this work. Special thanks too to Libby and Suzie for their support and much valued friendship throughout my PhD and before.

Finally, I would like to thank to my family, Sinéad, Peter, Rory and especially my parents, for their love and support throughout my education and life. I must also acknowledge my debt to my father for all that help with French and maths, which has made such a difference, from the age of four until now! Last but certainly not least thanks to Pepe, for coming to Ireland, for his love and care, and for putting up, in such a charming way, with one stressed-out gringuita.



I gratefully acknowledge the financial support provided by the TCD Broad Curriculum Fellowship initiative.

DEIRDRE HOGAN

*University of Dublin, Trinity College*  
*May 2007*

**On Coordination Disambiguation in a Generative  
Parsing Model, with Memory-Based Techniques for  
Parameter Estimation**

Publication No. \_\_\_\_\_

Deirdre Hogan, Ph.D.

University of Dublin, Trinity College, 2007

Supervisors: Saturnino Luz and Pádraig Cunningham

This thesis is concerned with improving existing generative history-based probability models' treatment of noun phrase coordination and the development of memory-based techniques for model parameter estimation.

Lexicalised generative history-based parsing models have proven to be highly successful at robust and accurate parsing. Although such models already achieve impressive overall accuracy results there is nevertheless potential for improvement, particularly in areas of difficulty for such parsers, such as coordination disambiguation and, more generally, parameter estimation from sparse data.

Though coordination has long been known as an area of difficulty for natural language parsing, coordination ambiguity is nevertheless a little studied area. Our aim is to increase understanding of coordination ambiguity in generative history-based parsing models. We seek to find ways of improving the model's handling of noun phrase coordination without removing coordination from the parsing framework. As well as reducing noise in the data, we look at modelling two main sources of information for disambiguation: symmetry in conjunct structure, and the likelihood of one lexical head being conjoined with another. The latter step involves extending the modelling of coordinate heads to include those found in base noun phrases and improving parameter estimation by incorporating data from the BNC and using a word graph and a measure of word similarity to decrease data sparsity. We also alter the head-finding rules for base noun phrases so that the lexical item chosen to head the entire phrase more closely resembles the head chosen for other types of coordinate noun phrase.

A difficulty in improving probabilistic generative models is how to incorporate into the probability model features that will capture information in the data important for disambiguation decisions within the limitations of feature selection in history-based models. In addition, adding new features to the model increases the risk of the sparse data problem and smoothing techniques which can overcome the sparse data problem when estimating the model parameters are important. We study the use of memory-

based techniques for parameter estimation and demonstrate that they are effective for parameter estimation in a lexicalised generative parsing model, allowing for flexible feature selection, good smoothing of data, and can achieve state-of-the-art results for accuracy.



# Contents

Acknowledgments	v
Abstract	vii
List of Tables	xiv
List of Figures	xvi
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Thesis Outline . . . . .	1
1.1.1 Coordination Disambiguation . . . . .	4
1.2 Generative Lexicalised History-based Parsing models . . . . .	8
1.2.1 Generative . . . . .	8
1.2.2 Lexicalised . . . . .	10
1.2.3 History-Based . . . . .	11
1.2.4 Markovised . . . . .	12
1.3 Coordination in the Baseline Model . . . . .	12
1.4 Parameter Estimation . . . . .	14
1.4.1 Linear Interpolation and Witten-Bell Estimation . . . . .	14
1.4.2 $k$ -Nearest Neighbour Parameter Estimation . . . . .	16
1.4.3 Similarity for Smoothing . . . . .	16
1.5 Chapter by Chapter Guide to this Thesis . . . . .	17
<b>Chapter 2 Previous Work</b>	<b>20</b>
2.1 Introduction . . . . .	20
2.2 Developments in History-Based Statistical Parsing . . . . .	21

2.2.1	[Magerman and Marcus, 1991, Magerman and Weir, 1992, Black et al., 1992] . . . . .	21
2.2.2	[Jelinek et al., 1994, Magerman, 1994, 1995] . . . . .	22
2.2.3	[Collins, 1996, 1997, 1999] . . . . .	23
2.2.4	[Charniak, 1996b, 1997, 2000] . . . . .	24
2.2.5	[Ratnaparkhi, 1997, 1998a] . . . . .	25
2.2.6	Henderson [2003] . . . . .	26
2.2.7	Investigations into the Importance of Lexical Statistics . . . . .	26
2.3	Ranking Algorithms . . . . .	27
2.3.1	$n$ -best lists . . . . .	30
2.4	Memory-based Learning and Natural Language Processing . . . . .	31
2.4.1	Advantages of Local Learning for Natural Language Learning Tasks . . . . .	31
2.4.2	Memory-Based Parsing . . . . .	32
2.5	Similarity for Smoothing . . . . .	34
2.5.1	$k$ -NN for Smoothing . . . . .	34
2.5.2	Cooccurrence Smoothing . . . . .	35
2.6	Previous Work on Coordination Ambiguity Resolution . . . . .	37

**Chapter 3 Memory-Based Parameter Estimation** . . . . . 41

3.1	Introduction . . . . .	41
3.2	Motivation . . . . .	41
3.3	The Baseline Model . . . . .	42
3.4	The Memory-Based Model . . . . .	43
3.4.1	Constraint Features for Training Set Restriction . . . . .	44
3.4.2	Smoothing . . . . .	45
3.4.3	Lexical Statistics . . . . .	46
3.5	Experiments . . . . .	47
3.5.1	Experimental Set up . . . . .	47
3.5.2	Experimental Details . . . . .	47
3.6	Results . . . . .	50
3.7	Computational Costs . . . . .	52
3.8	Relation to Previous Work . . . . .	52

3.9	Conclusion . . . . .	53
<b>Chapter 4</b>	<b>Conjoined Lexical Head Nouns</b>	<b>55</b>
4.1	Introduction . . . . .	55
4.2	Measures of Word Similarity . . . . .	55
4.2.1	Similarity based on Coordination Cooccurrences . . . . .	57
4.2.2	WordNet-Based Similarity Measures . . . . .	58
4.2.3	Empirical Evaluation of Similarity Measures . . . . .	59
4.2.4	Discussion . . . . .	61
4.3	Modelling Coordinate Head Words . . . . .	62
4.3.1	Extending $P_{coordWord}$ to Coordinate NPBs . . . . .	64
4.3.2	Estimating the $P_{coordWord}$ Parameter Class from a Coordination Word Graph . . . . .	64
4.4	Relation to Previous Work . . . . .	67
4.5	Summary . . . . .	69
<b>Chapter 5</b>	<b>Parallelism Across Conjuncts</b>	<b>71</b>
5.1	Introduction . . . . .	71
5.2	Empirical Measurements of Parallelism . . . . .	71
5.2.1	Methodology . . . . .	72
5.2.2	Results . . . . .	76
5.3	Modelling Symmetry in Conjuncts . . . . .	79
5.4	Relation to Previous Work . . . . .	80
5.5	Summary . . . . .	81
<b>Chapter 6</b>	<b>Noun Phrase Coordination Error Analysis</b>	<b>82</b>
6.1	Introduction . . . . .	82
6.2	Bracketing Guidelines for the Penn Treebank and Inconsistencies in WSJ Coordinate Noun Phrase Annotation . . . . .	83
6.3	NPB Head-Finding Rules . . . . .	85
6.3.1	Modifying the NPB Head-Finding Rules . . . . .	88
6.4	Relation to Previous Work . . . . .	90
6.5	Summary . . . . .	90

<b>Chapter 7</b>	<b>Experimental Evaluation - Coordination</b>	<b>91</b>
7.1	Introduction . . . . .	91
7.2	Experimental Evaluation . . . . .	91
7.3	Experimental Details and Results . . . . .	93
7.3.1	Eliminating Noisy Data . . . . .	93
7.3.2	Modelling Symmetry in Conjunct Structure . . . . .	94
7.3.3	NPB Head-Finding Rule and New Features for NPBs . . . . .	94
7.3.4	Modelling Conjoined Head Nouns . . . . .	95
7.3.5	Results . . . . .	96
7.4	Discussion . . . . .	98
7.5	Summary . . . . .	99
<b>Chapter 8</b>	<b>Conclusions and Future Work</b>	<b>101</b>
8.1	Summary . . . . .	101
8.1.1	Memory-Based Parameter Estimation . . . . .	102
8.1.2	Noun Phrase Coordination Disambiguation . . . . .	103
8.2	Future Work . . . . .	104
8.2.1	Feature Weighting . . . . .	104
8.2.2	Constraint Features . . . . .	104
8.2.3	Smoothing . . . . .	105
8.2.4	Word Graph and Similarity Function . . . . .	105
8.2.5	Modelling Dependencies Across CCs . . . . .	106
8.2.6	Cleaning Noisy Data . . . . .	107
8.2.7	Second-Pass Parsing and Discriminative Reranking . . . . .	107



# List of Tables

3.1	The parameter class for generating $C_h$ , the non-terminal label of the head child node. $C_p$ is the parent non-terminal label, $w_p$ and $t_p$ its head word and part-of-speech respectively, and $t_{gp}$ is the POS tag of the grandparent node. . . . .	49
3.2	The parameter classes for the generation of modifier nodes. The notation is that used throughout the thesis. <i>dir</i> is a flag which indicates whether the modifier being generated is to the left or the right of the head child. <i>dist</i> is the distance metric used in the Collins parser. $t_{i-1}$ and $t_{i-2}$ are the POS tags for the previous two generated nodes. $C_{gp}$ is the grandparent non-terminal label. . . . .	49
3.3	The parameter classes used only when $C_p = \text{NPB}$ . The notation is that used throughout the thesis. In addition, $C_{ggp}$ and $C_{gggp}$ are the great- and great-great- grandparent non-terminal labels respectively. $C_{i-2}, w_{i-2}$ and $C_{i-3}, w_{i-3}$ are the non-terminal labels and head words of the second and third previously generated nodes. . . . .	49
3.4	Results for sentences of less than or equal to 40 words, from section 23 of the Penn treebank. LP/LR =Labelled Precision/Recall. CBs = the average number of Crossing Brackets per sentence. 0 CBs, 2 CBs are the percentage of sentences with 0 or $\leq 2$ crossing brackets respectively. WB Baseline is our baseline emulation of Model 1 when tested on the output of the Bikel n-best parser. CO99 M1 and M2 are [Collins, 1999] Models 1 and 2 respectively. Bikel 1-best is [Bikel, 2004a]. $k$ -NN is our final $k$ -NN model. . . . .	50

4.1	Summary of the 9 different word similarity measures to be evaluated empirically on WSJ cooccurrence data. . . . .	59
4.2	Summary statistics for 9 different word similarity measures (plus one random measure): $n_{coord}$ and $n_{nonCoord}$ are the sample sizes for the coordinate and non-coordinate noun pairs samples, respectively; $\bar{x}_{coord}$ , $SD_{coord}$ and $\bar{x}_{nonCoord}$ , $SD_{nonCoord}$ are the sample means and standard deviations for the two sets. The 95% CI column shows the 95% confidence interval for the difference between the two sample means. The p-value is for a Welch two sample two-sided t-test. . . . .	60
5.1	Nodes aligned at level 1 for the trees in Figure 5.2 . . . . .	73
5.2	Contingency table for the head child non-terminal label <i>TO</i> at conjunct depth 1. . . . .	74
5.3	Percentage Match(%M) of head event labels $C_h$ in right-of-head conjuncts with the corresponding label in the head conjunct, grouped by Depth. Percentage match for head conjunct nodes collected in both a left-to-right (L-R) traversal and head-first (H-F) traversal are shown. . . . .	77
5.4	Percentage Match(%M) of $C_i$ and $t_i$ labels of dependent events in right-of-head conjuncts with the head conjunct, grouped by depth. Percentage match for head conjunct nodes collected in both a left-to-right (L-R) traversal and head-first (H-F) traversal are shown. The total number of dependent events ( $ DepEvents $ ) in post- <i>CC</i> conjuncts for each level is displayed. . . . .	78
7.1	Results on the Validation Set. 1064 coordinate noun phrases dependencies. In the significance column > means at level .05 and $\gg$ means at level .005, for McNemar's test of significance. Results are cumulative. . . . .	93
7.2	Results for Section 23. 416 coordinate noun phrase dependencies . . . . .	96

# List of Figures

1.1	Tree 1. The correct noun phrase parse. Tree 2. The incorrect parse for the noun phrase. . . . .	5
1.2	Tree 1. The correct noun phrase parse. Tree 2. The incorrect parse for the noun phrase. . . . .	6
1.3	The basic form of a coordinated phrase. <i>coord</i> refers to the coordination flag. . . . .	13
2.1	Parser and Reranking F-score Results Comparison on Section 23 of the WSJ . . . . .	27
4.1	Tree 1. The correct noun phrase parse. Tree 2. The incorrect parse for the noun phrase. . . . .	56
4.2	Graph of coordinations extracted from the BNC. . . . .	65
5.1	Example of symmetry in conjunct structure in a lexicalised subtree. . .	72
5.2	Trees that contain conjuncts with non-isomorphic structure. . . . .	73
5.3	Prior and posterior (positive adaption) probabilities for head child non-terminal labels at conjunct depth 1 . . . . .	75
5.4	Prior and posterior (positive adaption) probabilities for head child non-terminal labels at conjunct depth 5 . . . . .	75
5.5	Prior and posterior (positive adaption) probabilities for modifier POS labels at conjunct depth 1 . . . . .	76
6.1	Correct Parse tree bracketing according to the Penn Guidelines . . . .	85

6.2 Tree 1: The correct oracle coordinate NP. Tree 2: The incorrect coordinate NP returned by the baseline model. Tree 3: The oracle tree aligned with Tree 2. . . . . 87





# Chapter 1

## Introduction

### 1.1 Thesis Outline

Lexicalised generative history-based parsing models have proven to be highly successful at robust and accurate parsing [Collins, 1999, Charniak, 2000]. Developed from relatively simple Probabilistic Context Free Grammar (PCFG) models [Booth and Thompson, 1973], they are now highly complex models which weaken the independence assumptions of PCFGs by using information from previously generated parse structure to help predict the remaining structure of the parse tree. Although these models already achieve impressive overall accuracy results there is nevertheless potential for improvement. This is particularly true in areas of difficulty for such parsers, such as, for example, coordination disambiguation, prepositional phrase (PP) attachment, or, more generally, the estimation of parameters from sparse data.

Probabilistic parsing of natural language can be broken down into three main components: defining a probability model, estimating the parameters of the model, and efficiently searching for the most probable parse from the space of all possible parses for the sentence. The work in this thesis is concerned with improving parameter estimation in a generative model by using memory-based techniques as well as improving the model's handling of coordination disambiguation and so lies within the first two areas.

A difficulty in improving probabilistic history-based models is how to incorporate into the probability model features that will capture information in the data important

for disambiguation decisions within the limitations of feature selection in history-based models. In addition, adding new features to the model increases the sparse data problem, one of the core difficulties in empirical NLP. Increasing the number of conditioning features when predicting future structure can improve accuracy as the model has more information on which to base its prediction. However, increasing the number of conditioning features increases the number of parameters in the model, spreading the data over more specific events, and often there is simply not enough training data to be able to accurately estimate the probabilities of events. This is especially true when dealing with features that involve individual words, features which nevertheless are important as individual words tend to have good discriminating power. The method of estimating the local distributions, therefore, plays a very important role in building a good model.

In this thesis we examine the use of memory-based techniques for parameter estimation. Specifically, we use  $k$ -nearest neighbour ( $k$ -NN) for smoothing model parameters. Essentially, this technique involves basing the estimation of a particular parameter, or query instance, on the distribution of the class variable (or future) over the set of  $k$  instances from the training data that are most similar to the query instance.  $k$  is typically very large in probability estimation, compared to when  $k$ -NN is used for classification. Instances selected are weighted according to their similarity to the query instance, so that instances from memory that are more similar to the query instance will be given more weight in the prediction of the class value.

Memory-based learning is distinguished from other machine learning algorithms in that it delays generalising beyond the training data until it must classify, or assign a probability to, each new query instance. This sort of lazy learning avoids committing to a single global approximation at training time but instead implicitly represents the target function by a combination of many local approximations, which take into consideration the query instance when deciding how to generalise. The specific advantages of memory-based learning - the ability to model complex target functions by a collection of local approximations and the fact that memory-based learning does not abstract away from low frequency data - suggest, considering the irregularities and small sets of exceptions in natural language, that memory-based learning algorithms should lend themselves well to natural language learning [Daelemans et al., 1999a, Daelemans and van den Bosch, 2005].

We show that memory-based learning works well for parameter estimation in a

generative parsing model.  $k$ -NN is a very simple, but effective method, allows for flexible feature selection and achieves state-of-the-art performance in accuracy.

We carry out our experiments within the framework of generative parse reranking. We begin by describing a generative probabilistic model for parsing, based on Model 1 of Collins [1999], which re-estimates the probability of each parse generated by an initial base parser (Bikel [2004a]’s implementation of the Collins parser) using memory-based techniques to estimate local probabilities. We achieve an  $f$ -score of 89.4% for sentences  $\leq 40$  words on section 23 of the Penn Wall Street Journal (WSJ) Treebank [Bies et al., 1995], which represents a significant increase over our baseline parser and the Collins parser. Although the model effectively reranks the top- $n$  parses output from the base parser, insofar as it is generative the approach is more similar to a second-pass of a generative parser than to the discriminative reranking approaches of [Collins, 2000, Collins and Duffy, 2002, Shen et al., 2003, Henderson, 2004, Charniak and Johnson, 2005, Koo and Collins, 2005].

Discriminative approaches to parse reranking have recently become popular, motivated to a large extent by the flexibility of discriminative techniques in terms of feature selection compared to history-based models. Discriminative reranking approaches can choose features which incorporate arbitrary aspects of the whole parse tree structure, whereas in history-based models the choice of conditioning features when predicting parse structure is limited to structure that has already been determined in the derivation of the tree. Although discriminative reranking tends to improve on the performance generative models, there remain relatively small differences in accuracy between generative and discriminative models when tested on the Penn Wall Street Journal Treebank, despite the more restricted choice of features possible in history-based models.

Generative parsing models and discriminative rerankers are not competing strategies to parsing, however, but are complementary. Discriminative rerankers rely on history-based models to generate the  $n$ -best list of parses. In addition, the probabilities generated by a base generative model for each of the parses in an initial  $n$ -best parse ranking play an important role in several discriminative approaches to parse reranking [Collins, 2000, Collins and Duffy, 2002, Shen et al., 2003, Henderson, 2004, Charniak and Johnson, 2005, Koo and Collins, 2005]. Thus any improvements in the base generative model are likely to improve the discriminative rerankers. Generative



history-based reranking models have also an advantage in that they can be applied to the full output of the base parser and not just the  $n$ -best list to which discriminative rerankers are limited. This is because, unlike discriminative reranking approaches, history-based models can take advantage of a packed representation of trees and can use dynamic programming to search for the most probable tree according to the model.

The remainder of the thesis involves taking the memory-based model as the baseline model and working on improving the area in which the model performs worst: coordination disambiguation.

### 1.1.1 Coordination Disambiguation

As an example of the coordination disambiguation task, take the phrase *people of all ages and all classes*. The coordinating conjunction (CC) *and* and the noun phrase *all classes* could attach to the noun phrase *all ages*, as illustrated in Tree 1, Figure 1.1. Alternatively, *all classes* could be incorrectly conjoined to the noun phrase *people of all ages* as in Tree 2, Figure 1.1. This problem of whether to attach low (Tree 1) or attach high (Tree 2) is a common source of error in coordinate noun phrase disambiguation.

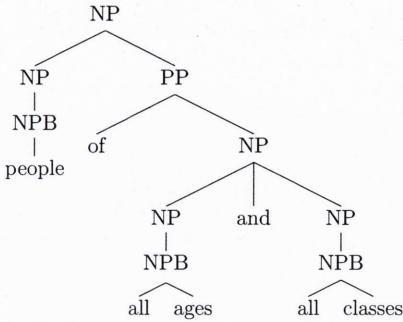
Another common source of disambiguation error is illustrated with the alternative bracketing of the slightly modified phrase *people of all ages and classes* shown in Figure 1.2. Here, the problem is whether *all* modifies both *ages* and *classes* as in Tree 1, Figure 1.2, or whether *all* modifies *ages* but not *classes* as in Tree 2, Figure 1.2.

Although there has been a substantial body of work on other areas of difficulty for parsers, such as PP-attachment, coordination ambiguity is a relatively little studied area. One reason for this could be that dependencies involving PP-attachment tend to occur much more often than coordination constructions. Thus, improving PP-attachment has perhaps greater potential to improve overall parser performance. The correct bracketing of coordination constructions, however, remains one of the most difficult problems for natural language parsers and parsers often perform worse at coordination disambiguation than PP-attachment. In the Collins parser and our emulation of his parsing model, dependencies involving coordination achieve by far the worst performance of all dependencies.<sup>1</sup>

---

<sup>1</sup>For example, in [Collins, 1999] an error analysis shows that although dependencies involving coordination conjunctions achieved  $f$ -scores as low as 61.8%, the lowest of all dependency types, compared to an  $f$ -score of 81.9% for PP modification, coordination accounts for only 1.9% of all

1.



2.

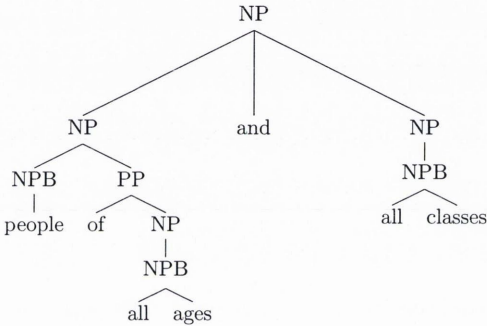


Figure 1.1: Tree 1. The correct noun phrase parse. Tree 2. The incorrect parse for the noun phrase.

As with PP attachment, most previous attempts at tackling coordination as a sub-problem of parsing have treated it as a separate task to parsing. It is not always obvious, however, how to integrate the methods proposed for disambiguation into existing parsing models, which presumably is the end goal of any work on PP or coordination disambiguation. Therefore we approach coordination disambiguation, not as a separate task, but integrated within the framework of a generative parsing model. Our aim is to increase understanding of coordination ambiguity in generative history-based parsing models and to improve the ability of a generative history-based parsing model to make the correct coordination decisions in the context of parse reranking. As noun

---

dependencies whereas PP-attachments accounts for 11.2%

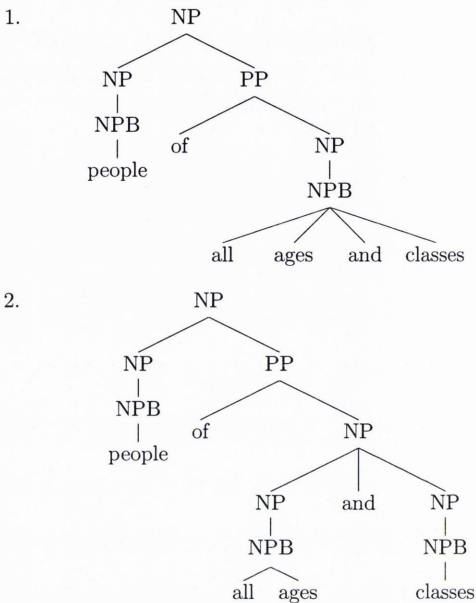


Figure 1.2: Tree 1. The correct noun phrase parse. Tree 2. The incorrect parse for the noun phrase.

phrase (NP) coordination accounts for over 50% of coordination dependency error in our baseline model we focus primarily on NP coordination.

We examine some of the types of error made in noun phrase coordination, showing how Penn Treebank data for NP coordination is particularly noisy and how inconsistencies in the Penn Treebank WSJ annotation of coordinate NPs negatively affect parser performance. We also show how the different head-finding rules for noun phrases and non-recursive noun phrases (base NPs) affect disambiguation, suggesting slightly modified head-finding rules for base NPs.

We look at the dependencies between two head nouns in coordinate noun phrases. We first introduce our distributional word similarity measure and compare it with several existing measures of word similarity, testing whether the various measures can detect similarity between the head nouns in coordinate noun phrases. We then concen-



trate on modelling the likelihood of two nouns conjoining, designing a new parameter class<sup>2</sup> for use in both coordinate noun phrases and coordinate base noun phrases. In the estimation of this parameter class, data from the unlabelled British National Corpus (BNC)[Burnard, 1995] are used in addition to WSJ data. We use a word graph to store the training data and explore variations of  $k$ -nearest neighbour which incorporate our measure of word similarity in the estimation of parameters in order to reduce data sparseness.

There is often a considerable bias toward symmetry in the syntactic structure of two conjuncts and most previous work on coordination disambiguation has attempted to take advantage of this. We give empirical measurements of the extent to which parallelism in the syntactic structure of conjuncts exists and then design new parameter classes for the generative model which attempt to capture the parallelism effect and thus allow the model to learn a bias toward symmetry in conjuncts.

The various changes to the baseline model in the handling of coordination result in a rise in NP coordination dependency  $f$ -score from 69.9% to 73.9%, which represents a relative reduction in  $f$ -score error of 13%.

We now summarise the contributions made in this thesis:

## 1. Parameter Estimation - Combating Data Sparseness

- We demonstrate that memory-based models, based on varieties of the  $k$ -nearest neighbour algorithm, are effective for parameter estimation in a lexicalised generative parsing model, allowing for flexible feature selection and good smoothing of data, and can achieve state-of-the-art results for accuracy.
- We introduce a novel technique for the estimation of certain types of bilingual statistics, which makes use of both labelled and unlabelled data and incorporates, for the first time, a measure of word similarity into a generative lexicalised parsing model.

## 2. Coordinate Noun Phrase Disambiguation

---

<sup>2</sup>We refer to a model space, such as, for example, the bigram model  $P(w_i|w_{i-1})$ , as a parameter class. A particular parameter from that parameter class might be  $P(w_i = \text{cat}|w_{i-1} = \text{the})$ .

- We investigate some of the causes for the errors in coordinate noun phrase disambiguation, showing that the data used for such parsers are particularly noisy with regard to NP coordination. We also demonstrate how head-finding rules can negatively affect disambiguation.
- We give an empirical analysis of noun phrase coordination in the data - focusing on two salient characteristics of noun phrase coordination: symmetry in conjunct structure and word similarity for coordinate head nouns.
- Based on the study of training data and parser errors we develop techniques for improving the model's ability to disambiguate coordinate structure, including altering the parameterisation of the model and improving parameter estimation.

An early version of the work on memory-based parameter estimation for generative parsing was published in [Hogan, 2005]. Hogan [2007a] describes some of the work on coordinate noun phrase disambiguation reported in this thesis and Hogan [2007b] reports on the empirical measurements of lexical similarity in noun phrase conjuncts presented in Chapter 4.

The remainder of this chapter is organised as follows: In Section 1.2 we outline the generative history-based parsing model adopted in this thesis, introducing the notation that will be subsequently used throughout the thesis. Then, in Section 1.3, we give a brief overview of how coordination is handled in the Collins parsing model - our baseline model. In Section 1.4 we outline the parameter estimation techniques used in this thesis: linear interpolation and the Witten-Bell estimation of the baseline model and  $k$ -nearest neighbour methods. Finally, Section 1.5 gives a chapter by chapter guide to the rest of the thesis.

## 1.2 Generative Lexicalised History-based Parsing models

### 1.2.1 Generative

Generative parsing models estimate the joint probability,  $P(t, s)$ , for each candidate parse tree  $t$  of a sentence  $s$ , where  $s \in S$ ,  $t \in T$ , and  $S$  is the set of all sentences in the

language and  $T$  the set of parse trees. Each tree in  $T$  has a member of  $S$  as its yield (i.e. its sequence of leaf nodes).

Generative probability models define a joint probability distribution,  $P(t, s)$  over the space of all possible sentence/parse tree pairs, which satisfies the constraint:

$$\sum_{t \in T, s \in S} P(t, s) = 1 \quad (1.1)$$

As probabilities are for the entire language, it is possible to find the overall probability of a sentence:

$$P(s) = \sum_{t \in T} P(t, s) \quad (1.2)$$

Generative parsing models estimate  $P(t|s)$  indirectly by making the observation that maximising  $P(t, s)$  is equivalent to maximising  $P(t|s)$ . The most likely parse tree,  $\hat{t}$ , is given by:

$$\hat{t} = \operatorname{argmax}_{t \in T} P(t|s) = \operatorname{argmax}_{t \in T} \frac{P(t, s)}{P(s)} = \operatorname{argmax}_{t \in T} P(t, s) \quad (1.3)$$

(In (1.3)  $P(s)$  is constant so maximising  $\frac{P(t, s)}{P(s)}$  is equivalent to maximising  $P(t, s)$ ). The joint probability  $P(t, s)$  is simply  $P(t)$  where the yield of  $t$  is equal to  $s$ , and 0 otherwise. Thus, from the space of all candidate parses for a particular sentence, generative parsers choose the parse tree that maximises the probability  $P(t)$ .

The probability of a tree is calculated as the product of all the rewrite rules from which the tree is derived. In a PCFG, for a tree derived by  $n$  applications of context-free rewrite rules  $LHS_i \rightarrow RHS_i$ ,<sup>3</sup>  $1 \leq i \leq n$ ,

$$P(t) = \prod_{i=1..n} P(RHS_i | LHS_i) \quad (1.4)$$

In PCFGs the context-free rewrite rules are so called because they are independent of surrounding context in the tree - that is the probability of a rule expansion is independent of where the rule occurs in the tree. The probability of a rewrite rule is estimated using relative frequency estimates:

---

<sup>3</sup>LeftHandSide (LHS)  $\rightarrow$  RightHandSide (RHS)

$$P(RHS_i|LHS_i) = \frac{\text{count}(LHS_i \rightarrow RHS_i)}{\text{count}(LHS_i)} \quad (1.5)$$

where  $\text{count}(LHS_i \rightarrow RHS_i)$  and  $\text{count}(LHS_i)$  return the frequency of  $LHS_i \rightarrow RHS_i$  and  $LHS_i$  in the corpus respectively. This is the maximum likelihood estimate (MLE) (see [Collins, 1999, p. 40] for proof on why, in this case, the relative frequency estimate is the maximum likelihood estimate, and also for a fuller description of the generative parsing model than is given here).

### 1.2.2 Lexicalised

Lexicalisation tends to improve parser accuracy because it allows the parser to use crucial information about the words in the sentence when disambiguating the syntactic structures of that sentence (see, for example, early work on lexical statistics for resolving syntactic ambiguity in [Hindle and Rooth, 1991]).

A PCFG can be lexicalised by associating a word,  $w$ , and also a part-of-speech (POS) tag,  $t$ , with each non-terminal in the tree. The key idea is that each constituent has a ‘head’ which is its most important lexical item.

An unlexicalised PCFG rewrite rule can be written as:

$$C_p \rightarrow C_{l_n} \dots C_{l_1} C_h C_{r_1} \dots C_{r_m} \quad (1.6)$$

where, on the left hand side of the rule,  $C_p$  is the parent constituent label. The right hand side of the rule consists of the children of  $C_p$ : a sequence of  $n$  constituents to the left of the head child constituent (left modifiers), followed by the head constituent,  $C_h$ , followed by the  $m$  constituents to the right of the head constituent (right modifiers).

The lexicalised version of (1.6) is:

$$C_p(w_p, t_p) \rightarrow C_{l_n}(w_{l_n}, t_{l_n}) \dots C_{l_1}(w_{l_1}, t_{l_1}) C_h(w_p, t_p) C_{r_1}(w_{r_1}, t_{r_1}) \dots C_{r_m}(w_{r_m}, t_{r_m}) \quad (1.7)$$

where each constituent is associated with its head word,  $w_i$  and head word POS tag  $t_i$ . Note that the head word and POS tag of  $C_h$  - the head child - are inherited from the parent constituent.

The introduction of lexicalisation vastly increases the number of rules in the grammar and makes direct estimation of constituent expansion rules unfeasible because of sparse data problems. Using the chain rule of probabilities the probability of a



rule is decomposed into the product of more tractable probabilities and independence assumptions are made to reduce the number of parameters in the model.

### 1.2.3 History-Based

In order to incorporate richer context in the probability model, in an attempt to overcome the structural weaknesses inherent in the independence assumptions of probabilistic context-free grammars, history-based models [Black et al., 1992] were developed. In history-based models the probability of a derivation  $D$  of a parse tree is the product of the probabilities of each step in the derivation of the tree. For example, for PCFGs, each step, or decision, in the derivation of the tree is an application of a rewrite rule. Unlike PCFGs, however, in history-based models the probability of a step  $d_i$  in the construction of a tree is conditioned on potentially all structure that has already been determined in the derivation of the tree. For a tree derived by a sequence of  $n$  decisions:

$$P(D) = \prod_{i=1..n} P(d_i | d_1, \dots, d_{i-1}) \quad (1.8)$$

The sequence of previous decisions  $d_1, \dots, d_{i-1}$  is referred to as the history of  $d_i$ . In practice it is not practical to condition on the entire history as this would lead to a vast number of parameters. Instead a history mapping function  $\Phi$  maps the history to a finite set of history contexts, so that:

$$P(D) = \prod_{i=1..n} P(d_i | \Phi(d_1, \dots, d_{i-1})) \quad (1.9)$$

PCFGs are a special case of history-based model, where the history of a rule expansion is taken simply to be the non-terminal label of the node being expanded.

How a tree is derived is important as it affects the choice of conditioning features at each step in the generation of the tree. In PCFGs and in the models used in this thesis there is a one-to-one mapping between a tree and its derivation. Thus  $P(D) = P(t)$ . This is not always necessarily the case however. Where multiple derivations are possible for a tree, the probability of a tree is the sum of the probabilities of each possible derivation (as with, for example, the parsing models of [Magerman, 1994, Bod,

1998]):

$$P(t) = \sum_D P(D) \quad (1.10)$$

### 1.2.4 Markovised

We use a generative model for parsing following the lexicalised history-based model of Collins [1999] where the grammar rules are Markovised.

Take the lexicalised rule in (1.7). In a Markov grammar, instead of generating the right-hand-side in one step, the generation process is broken down into three main steps: first the head child  $C_h$  is generated, then  $C_{l1}(w_{l1}, t_{l1})$  through to  $C_{ln}(w_{ln}, t_{ln})$ , then  $C_{r1}(w_{r1}, t_{r1})$  through to  $C_{rm}(w_{rm}, t_{rm})$ . At each step, the probability of generating a particular child node can be conditioned on the children which have already been generated. In a first order Markov grammar a modifier node is conditioned on the previously generated node (as well as the parent node). In an  $m^{th}$  order Markov grammar the node is conditioned on the  $m$  previously generated siblings (and parent node). The model also generates two special +STOP+ nonterminals as the leftmost ( $ln+1$ ) and rightmost ( $rm+1$ ) children of every parent. In a markovised grammar the generation of the +STOP+ nonterminals is necessary if the model is to sum to 1, due to the fact that constituents have a variable number of children. See [Collins, 1999, p. 46] for a discussion on the importance of generating the +STOP+ symbols.

An advantage of using a Markov grammar is that breaking down the generation of the child nodes of a constituent into a series of steps helps combat data sparseness because it makes it possible to generate rules which have not occurred in the training data.

The term vertical markovisation is sometimes used when information from previously generated ancestor nodes is used as part of the local history in a parameter class.

## 1.3 Coordination in the Baseline Model

In this section we give a brief outline of the handling of coordination in the Collins parser - our baseline model. A more detailed description of coordination in the Collins model can be found in Bikel [2004b].

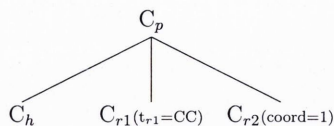


Figure 1.3: The basic form of a coordinated phrase. *coord* refers to the coordination flag.

In the Collins parsing model each node in a parse tree is annotated with a coordination flag, set to true if the node is conjoined to the head node of the phrase, and false otherwise. The head node of a coordinate phrase always precedes the coordination conjunction, and the coordinating conjunction followed by the second conjunct always occur to the right of the head conjunct. A coordinating conjunction node, followed by a conjunct, are generated together, unlike other constituents.

Take the tree fragment in Figure 1.3 where the POS tag  $t_{r_1}$  of the node following the head node is a coordinating conjunction. In such case the node following the CC node will have its coordination flag (*coord*) set to true. The CC node will not be generated as with other modifier nodes. Instead, node  $C_{r_2}$  is generated after the head node  $C_h$ . Then the CC node is generated via a special CC parameter class, conditioned on the two conjuncts  $C_h$  and  $C_{r_2}$

Coordination is handled differently for base noun phrases. A base, or non-recursive, noun phrase (NPB) as defined in [Collins, 1999], is a noun phrase which does not directly dominate another noun phrase, unless that noun phrase is possessive. For nodes in base noun phrases all coordination flags are set to false and modifier nodes are generated in the usual fashion with no special treatment of CC nodes. The reason for the different handling of coordination in base NPs is not stated in Collins' thesis. However, NPBs are treated differently to other constituent types in several ways. For all nodes, with the exception of NPBs, a modifier node to the left or right of the head node is always conditioned on the head node. In contrast, for base noun phrases the modifier node is conditioned on the previously generated node. As discussed in [Bikel, 2004b], the previously generated node in NPBs is treated as a head node for the purpose of conditioning and it is as a consequence of this that coordinate NPBs are not handled like other coordinate phrases.

The effect on disambiguation of the different handling of coordination in base NPs



is discussed in detail in §4.3.1, where we propose an alternative way of handling coordination in NPBs.

## 1.4 Parameter Estimation

One of the core difficulties in empirical NLP is the sparse data problem - often there is not enough data collected to enable accurate estimation of the probabilities of low-frequency events. This is particularly true when collecting data on events which include individual words. Due to the sparseness of NLP data, the method of estimating the local distributions plays a very important role in building a good model.

Data sparseness makes the maximum likelihood estimate for lexicalised rule probabilities unreliable, especially if we are to include more features from the history. With maximum likelihood estimation there will be a very large number of cases of rules which are given a zero probability, when in fact they should really have some non-zero probability. In this sense we can say that maximum likelihood estimation causes overfitting: all the probability mass is distributed over the cases we have already seen, with no probability mass left for a completely new case. Clearly there is a need to generalise, and this is what smoothing does in effect. This is a crucial step in natural language parsing where the aggregate probability of the unseen or low probability events can be significant.

As maximum likelihood estimation is known to be unreliable for low or zero counts, a variety of smoothing techniques has been developed to improve estimates. Chen and Goodman [1996] present a useful survey of smoothing techniques for language modelling as well as a comprehensive comparison of several techniques. Toutanova et al. [2003] also compare different estimation techniques, including a memory-based technique, in a HPSG parsing model. We focus here on the estimation techniques used in this thesis: that of the baseline Collins parser - a type of linear interpolation using Witten Bell smoothing - and then memory-based techniques.

### 1.4.1 Linear Interpolation and Witten-Bell Estimation

In linear interpolation the probability estimate of an event with history context  $X_i$  is interpolated with an estimate which has a more general context. The different histories

are sometimes referred to as backoff levels. The idea is that when there is insufficient data to estimate the more specific model, then the more general model might provide useful information.

In linear interpolation estimates for the probability of a class  $y$  (the future), given feature vector  $X_i$  (the history), where  $X_i$  is the history at backoff level  $i$ , are interpolated as follows:

$$P_{interp}(y|X_i) = \lambda_{X_i} P_{MLE}(y|X_i) + (1 - \lambda_{X_i}) P_{interp}(y|X_{i+1}) \quad (1.11)$$

$$P_{interp}(y|X_n) = P_{MLE}P(y|X_n) \quad (1.12)$$

where  $n$  is the number of backoff levels,  $0 \leq \lambda_{X_i} \leq 1$  and  $X_{i+1}$  is a feature vector less specific than  $X_i$  (i.e. with fewer features). That is, the smoothed model is defined recursively as a linear interpolation of the MLE of the more specific model and the smoothed estimate of the less specific model. The recursion ends by taking the smoothed estimate of the most general level of backoff to be the maximum likelihood estimate (alternatively, the uniform distribution could be taken as the final smoothed model).

One simple but effective method for calculating the  $\lambda$  values, which does not require extensive training, is the method used in [Collins, 1999], which was adapted from [Bikel et al., 1997] and the smoothing technique of [Witten and Bell, 1991].

$\lambda_{X_i}$  is defined in terms of  $count(X_i)$ , which is the number of times context  $X_i$  occurs in the corpus:

$$\lambda_{X_i} = \begin{cases} 0 & \text{if } count(X_i) = 0 \\ \frac{count(X_i)}{count(X_i) + C * D(X_i)} & \text{if } count(X_i) > 0 \end{cases}$$

where  $C$  is a constant which can be optimised using held-out data.  $D(X_i)$  is the diversity of the history  $X_i$ , that is the number of distinct outcomes that have been seen with context  $X_i$  in the training sample. We can interpret these calculations intuitively as follows: with probability  $\lambda_{X_i}$  we should use the higher order model and with probability  $1 - \lambda_{X_i}$ , the lower order model. If the particular context  $X_i$  has a high frequency of occurrence then a high value for  $\lambda_{X_i}$  is suitable because the higher-order distribution will be reliable. If the context has occurred very infrequently then a low value for  $\lambda_{X_i}$  is appropriate. If the context is highly diverse then we have less trust in the higher-order model and more in the lower-order one. This technique is sometimes

referred to as Witten-Bell interpolation.

### 1.4.2 $k$ -Nearest Neighbour Parameter Estimation

The probability of a class  $y$ , given feature vector  $X$ , can be estimated using the  $k$ -NN algorithm as follows:

$$P(y|X) = \frac{\sum_{X' \in N_k(X)} w(\Delta(X, X')) \delta(y, y')}{\sum_{X' \in N_k(X)} w(\Delta(X, X'))} \quad (1.13)$$

where  $\Delta(X, X')$  is the distance function between feature vectors.  $\delta(y, y')=1$  iff  $y = y'$ , otherwise 0.  $w(\Delta(X, X'))$  is the weight of neighbour  $X'$  of  $X$  where the weight is a function of the distance.  $N_k(X)$  is the set of  $k$ -nearest neighbours of  $X$ .

For categorical variables the distance function often used is the overlap metric which simply counts the number of mismatching feature values between instances  $X$  and  $X'$ :

$$\Delta(X, X') = \sum_{i=1}^n d(x_i, x'_i) \quad (1.14)$$

where:  $d(x_i, x'_i) = 0$  iff  $x_i = x'_i$  else 1.  $\Delta(X, X')$  is the distance between instances  $X$  and  $X'$ , represented by  $n$  features, and  $d$  is the distance per feature. In effect, the weighting function  $w(\Delta(X, X'))$  turns the distance into a measure of nearness, or similarity. A popular weighting function is the inverse distance function:

$$w(\Delta(X, X')) = \frac{1}{(\Delta(X, X') + 1)^m} \quad (1.15)$$

for some constant  $m$ .

### 1.4.3 Similarity for Smoothing

In this thesis, as well as using  $k$ -NN for parameter estimation as in (1.13), we use a variation which calculates the similarity function directly, rather than calculating the distance and then converting this to a similarity function:

$$P(y|X) = \frac{\sum_{X' \in N_k(X)} sim(X, X') \delta(y, y')}{\sum_{X' \in N_k(X)} sim(X, X')} \quad (1.16)$$

where  $\text{sim}(X, X')$  is a similarity score between instances  $X$  and  $X'$ .

If we group history samples together so that  $n_j$  is the history *type*  $X_j$  - that is,  $n_j$  refers to those history feature vectors which have the same values for each feature as  $X_j$ , and where  $\text{count}(n_j)$  is the number of history samples of type  $n_j$  in the data set. We can rewrite (1.16) as:

$$P(y|n_j) = \frac{\sum_{n_x \in N(n_j)} \text{sim}(n_j, n_x) \text{count}(y, n_x)}{\sum_{n_x \in N(n_j)} \text{sim}(n_j, n_x) \text{count}(n_x)} \quad (1.17)$$

where  $\text{count}(y, n_x)$  is the number of times future  $y$  occurs with history type  $n_x$  and  $\text{sim}(n_j, n_x)$  is a similarity score between types  $n_j$  and  $n_x$  and  $N(n_j)$  is the set of types in the neighbourhood of  $n_j$ . This is the form of our billexical estimate in §4.3.2 where we use a measure of similarity between words for smoothing.

## 1.5 Chapter by Chapter Guide to this Thesis

The work in this thesis began by replicating the state-of-the-art parser of [Collins, 1999] Model 1 and then altering this baseline model so that it used memory-based learning for parameter estimation. We then focused our attention on coordinate noun phrase disambiguation as this was the worst performing area of the parser. Our experiments on noun phrase coordination disambiguation began with an analysis of the errors produced by the memory-based model, leading us to look at inconsistencies in treebank annotation as a source of error. Noticing also a marked tendency toward parallelism across conjuncts, we then explored this area by first measuring empirically the extent of symmetry across conjuncts and, based on positive evidence of the same, we experimented with incorporating a bias toward symmetry in conjunct structure into the probability model. Our analysis of errors also led us to experiment with new head-finding rules for base noun phrases. We noticed too on inspection that many conjoined nouns appeared to be semantically similar and this motivated us to carry out experiments with different measures of similarity between conjoined nouns on the training set. In our final set of experiments we focused on modelling the likelihood of two nouns conjoining and reducing the sparsity for this parameter class, developing a similarity-based parameter estimation technique.

For the sake of coherence, readability and because of some cross-referencing issues



we do not always present the work in the thesis in chronological order of experiments carried out. The remainder of the thesis is arranged as follows.

*Chapter 2* begins with an overview of history-based approaches to statistical natural language parsing, followed by a brief look at recent approaches to discriminative reranking. This is followed by a summary of memory-based techniques in natural language processing that are most relevant to the work in this thesis. We also look at previous attempts to use similarity measures for smoothing bilinear probability estimates. Finally, we discuss previous approaches to coordination disambiguation.

*Chapter 3* presents our generative parsing model, with  $k$ -nearest neighbour parameter estimation. We describe a technique based on constraint features to reduce the size of the training set for each parameter class which helps both with accuracy and speed. We also show how we combine  $k$ -nearest neighbour with linear interpolation for bilinear statistics and present results which achieve state-of-the-art accuracy for generative models.

*Chapter 4* begins our focus on coordinate noun phrase disambiguation and is divided into two main parts. The first introduces our distributional word similarity measure and compares it with several existing measures of word similarity, testing whether the various measures can detect similarity between the head nouns in coordinate noun phrases. The second part of this chapter concentrates on modelling the likelihood of two nouns conjoining, designing a new parameter class for use in both coordinate noun phrases and coordinate base noun phrases. In the estimation of this parameter, data from the unlabelled British National Corpus are used in addition to WSJ data. We use a word graph to store the training data and incorporate our word similarity measure in the estimation of the parameter in order to reduce data sparseness.

*Chapter 5* begins with empirical measurements of the extent to which parallelism in the syntactic structure of conjuncts exists. We then design new parameter classes for the generative model which attempt to capture the parallelism effect and thus allow the model to learn a bias toward symmetry in conjuncts.

*Chapter 6* gives an analysis of some of the reasons for the baseline model's poor performance in the area of coordinate noun phrase disambiguation. We look at how inconsistencies in the Penn Treebank WSJ annotation of coordinate NPs negatively affects parser performance and also show how the different head-finding rules for NPs and NPBs affects disambiguation, suggesting a slightly modified head-finding rule for

base NPs.

*Chapter 7* shows how we evaluate the experiments on coordination disambiguation and gives the details of the experiments carried out. We outline the effects of each different experiment and discuss the results.

*Chapter 8* concludes on the work presented in the thesis and suggests avenues for further research.

# Chapter 2

## Previous Work

### 2.1 Introduction

In this chapter we summarise previous work most related to the work in this thesis. First, in Section 2.2, we trace the development of history-based parsing and the current state-of-the-art parsers, upon which our baseline model is based. Unless otherwise stated parser accuracy is reported on section 23 of the Penn WSJ treebank for sentences  $\leq 100$  words. In Section 2.3 we give a brief outline of recent work in discriminative reranking and  $n$ -best parsing. Section 2.4 moves on to memory-based learning of natural language. We summarise previous work on why memory-based learning is suited to natural language learning tasks and briefly outline previous work on parsing that comes under the broad category of memory-based learning. In Section 2.5 we turn to similarity for smoothing, first presenting previous work on smoothing with memory-based learning algorithm  $k$ -NN, and then looking at somewhat related work in nearest neighbour cooccurrence smoothing. Finally, in Section 2.6, we review previous work on coordination disambiguation.



## 2.2 Developments in History-Based Statistical Parsing

### 2.2.1 [Magerman and Marcus, 1991, Magerman and Weir, 1992, Black et al., 1992]

Some early work in overcoming the structural weakness inherent in the independence assumptions of the PCFG was that of [Magerman and Marcus, 1991, Magerman and Weir, 1992]. The Picky parser, and its predecessor Pearl, differed from previous work on probabilistic parsing in that a hand-crafted context-free grammar was modelled with *context-sensitive* conditional probabilities trained from a corpus. In the probabilistic model the probability of each parse tree  $T$  given a sentence  $S$  was defined as:

$$P(T|S) = \sum_{A \in T} P(A \rightarrow \alpha) | C \rightarrow \beta A \gamma, a_0, a_1, a_2 \quad (2.1)$$

where  $A$  is the non-terminal being expanded,  $C$  is the non-terminal node which immediately dominates  $A$ ,  $a_1$  is the part-of-speech of the left-most word of constituent  $A$ , and  $a_0$  and  $a_2$  are the POS tags of the words to the left and right of  $a_1$ , respectively.

Black et al. [1992] were the first to develop the concept of the history-based model which is distinguished from the context-free model in that for each constituent structure the conditioning was extended to look at potentially all previously built structure, rather than just the non-terminal being expanded as in PCFGs. As outlined in Section 1.2.3, in history-based models history is interpreted as any element of the parse tree which has already been determined and can include previous words, non-terminal labels, constituent structure, and any other linguistic information which is generated as part of the parse structure. In Black et al. [1992]'s generative model each constituent in the parse tree was associated with the following probability:

$$P(Syn, Sem, R, H_1, H_2 | Syn_p, Sem_p, R_p, I_{pc}, H_{1p}, H_{2p}) \quad (2.2)$$

where  $Syn$  and  $Sem$  are syntactic and semantic labels associated with the constituent,  $R$  is the constituent's re-write rule, and  $H_1$  and  $H_2$  are two lexical heads associated with the constituent. These are conditioned on the syntactic and semantic labels, re-write rule and lexical heads of the constituent's parent, as well as its index,  $I_{pc}$ , as a child

of  $R_p$ . This probability is decomposed into the product of five probabilities, of which all, bar one, are estimated using deleted interpolation. The other of the component probabilities are estimated using decision trees. The introduction of lexical information is noteworthy as most subsequent high-performing, broad coverage parsers use some degree of lexicalisation. Words were not represented as individual tokens but rather as bit strings via the clustering algorithm of [Brown et al., 1990].

### 2.2.2 [Jelinek et al., 1994, Magerman, 1994, 1995]

The parsing model developed by Jelinek et al. [1994] and extended in Magerman [1994] framed the natural language parsing task as one of treebank recognition. Unlike previous parsing models, which depended on carefully hand-crafted grammars, the model is presented with a treebank from which to learn and, given a sentence to parse, the task is to recognise the parse tree for the sentence that would be given it by a treebank annotator. The parsing model is a history-based conditional model. Unlike other history-based models, where a tree is associated with just one unique derivation, multiple derivations are possible and the probability of a tree is the sum of the probabilities for the various derivations of the tree.

Each decision made when building a particular parse derivation is conditional on decisions previously made within a certain window around the current node. Nodes in a parse tree are associated with various features and a parse tree is constructed by generating values for features of the tree nodes, bottom-up, one at a time, according to the distributions assigned by statistical models. The features for terminal nodes are the head word, head tag, and extension, where the extension feature connects the nodes in the tree and encode the tree's shaped. Internal nodes have the additional feature of the non-terminal label.

Four main statistical models are used in the construction of a tree: a POS tagging model, a non-terminal label model, an extension model, and a derivation model. Model parameters are estimated using statistical decision trees. As with all history-based models, where conditioning context is taken from the structure build so far, the derivation of a tree affects the conditioning features. The derivation model was introduced in order to allow more probability mass to be given to derivations in which the context available in the derivation of the tree suggested the correct parse, than

to derivations for which the local context at the various decisions was inconclusive or misleading. In the SPATTER parser of [Magerman, 1994] there was also a conjunction model in order to help predict the scope of conjunctions. Each node in the tree was associated with an additional boolean coordination flag, set to true for a particular constituent when the constituent is part of a conjoined phrase. As in [Black et al., 1992] words are represented as bit strings. The version of SPATTER described in [Magerman, 1994] is trained and tested on the IBM Computer Manuals domain. Magerman [1995] gives results of a version of the SPATTER parser, which does not include the derivation model, trained and tested on the WSJ corpus.

### 2.2.3 [Collins, 1996, 1997, 1999]

Collins [1996] presents a conditional parsing model where parse trees are lexicalised and represented as a set of head-modifier dependency relationships and a set of base noun phrases. Parameters are estimated using relative frequencies and a variation of the deleted interpolation method for smoothing described in [Jelinek, 1990]. Though a much simpler model than [Magerman, 1995] Collins' dependency model achieved a higher accuracy of 85.3%/85.7% labelled precision and recall on section 23. Mathematical shortcomings in the model, as well as some limitations due to parse representation, led to the improved generative model of [Collins, 1997], with some extra refinements reported in [Collins, 1999]. Collins [1997, 1999] presents three history-based generative models. The parsing model explored in this thesis is derived from Collins' Model 1. All three models are generative, lexicalised parsing models with first-order Markov grammar generation of nodes. Nodes in the parse tree are annotated with a coordination and punctuation flag, in addition to head word and head word part-of-speech information. For a more detailed description of the handling of coordination in the Collins generative model see §1.3. Model 2 adds a suffix 'C' to all non-terminals which are complements. In addition, a new parameter class for the generation of subcategorisation frames is introduced. Before the generation of a modifier non-terminal, its subcategorisation frame is generated, which is then used as a conditioning feature for the generation of the non-terminal label, head-word and so on. Finally, Model 3 integrates a probabilistic treatment of traces and Wh-movement into the parsing model, although this has little effect on the accuracy of the model. The parameter estimation



technique for all models is a simple but effective variation on linear interpolation, described in more detail in 1.4.1. The best results for sentences less than or equal to 100 words in length were achieved by Model 2: 88.3%/88.1% labelled precision/recall.

## 2.2.4 [Charniak, 1996b, 1997, 2000]

Charniak [1996b] also moved from depending on a hand-crafted grammar to relying solely on treebank parses. In [Charniak, 1996b] he describes treebank grammars, which are made up of CFG rules extracted from Penn treebank trees. Statistics are then collected from the treebank and associated with the rules for PCFG parsing. In his experiments he found that, contrary to common wisdom at the time, a probabilistic parser trained on treebank grammars outperformed those trained from hand-crafted grammars associated with treebank statistics. This work was developed considerably in Charniak [1997] by lexicalising the CFG rules and including more context in the probability estimates. In Charniak [1997]’s model there are two parameter classes. The probability of a rule expansion of the traditional PCFG is conditioned on increased contextual information, namely its head-word  $w_p$  and its parent  $C_{gp}$ :

$$P(C_p \rightarrow \alpha | w_p, C_p, C_{gp}) \quad (2.3)$$

The other parameter class is the probability of the head word of a constituent,  $w_i$ :

$$P(w_i | w_p, C_i, C_p) \quad (2.4)$$

conditioned on its non-terminal label  $C_i$  and the head word,  $w_p$ , of its parent node, with label  $C_p$ . Estimates were calculated using deleted interpolation, where the backoff weights were calculated as described in [Charniak, 1996a]. Interestingly, for both parameter classes the estimate, though initially conditioned on a word token, is backed-off to condition on a word class. For example, the second backoff term in the linear combination of the estimates in (2.4) is  $P(w_i | class_{w_p}, C_i, C_p)$ . Classes of words were induced by a clustering method similar to [Pereira et al., 1993]. Charniak concludes that although backing off to condition on word classes is worthwhile, statistics based on word classes alone, as in [Magerman, 1995], rather than on individual words, harms performance. At labelled precision/recall scores of 86.6%/86.7% on sentences less than or equal to 100 words in length on section 23 of the WSJ, Charniak’s model outperformed the earlier models of [Magerman, 1995, Collins, 1996].

In [Charniak, 2000], Charniak presents new developments in his generative parser which result in the highest parsing accuracy to date for generative parsers. The ‘maximum-entropy-inspired’ parser achieves labelled precision/recall results of 89.5%/89.6%. The most significant improvement in accuracy came from simply annotating nodes with the head word’s POS tag as well as the head word and then in the top-down derivation of the tree, generating the POS tag of a node before generating its head word, which is then conditioned on its POS tag. The effectiveness of this particular parameterisation was noted before in [Eisner, 1996, Collins, 1999]. Other improvements to the model came from increasing the local conditioning context in the model’s parameters to include, for example, such features as the label of the left sibling of the node being expanded. Parameters were estimated using a novel technique inspired by how features are handled in maximum-entropy estimation and which allowed increased flexibility when experimenting with different conditioning events. Following Collins [1999] the PCFG rules were markovised, with a third-order horizontal markovisation giving the best results.

Finally, a small but significant improvement came from explicitly marking noun and verb phrase coordinate structures. Unlike in [Magerman, 1995, Collins, 1999] where the conjuncts themselves are marked, in [Charniak, 2000] it seems that the parents of conjuncts are marked. A noun or verb phrase is marked as being a coordinate structure if it has two or more children of the same type (i.e. children which are noun phrases or verb phrases, respectively) as well as one or more of the constituents comma, CC, CONJP, and nothing else.

### 2.2.5 [Ratnaparkhi, 1997, 1998a]

Ratnaparkhi [1997, 1998a] describes a conditional history-based model, where each action taken by the parser is conditioned potentially on all actions taken thus far in the parse derivation. The probability of a parse is the product of the probabilities of the parser actions in the bottom-up generation of the parse tree. There is a one-to-one mapping between a parse tree and a parse derivation. There are four main parameter classes, based on parser actions, and which are estimated using maximum-entropy models. Some features make use of bilexical statistics and, for each feature which looks at pairs of head words there are one or more other features similar except that one or

more words will be omitted. This is a somewhat similar idea to linear interpolation which backs off to less specific context due to the sparsity of data. Ratnaparkhi's parsing model achieved accuracy of 87.5%/86.3% on the standard test set.

## 2.2.6 Henderson [2003]

Henderson [2003] presents another highly accurate generative history-based model, trained and tested on the WSJ. Rather than choosing by hand the conditioning features for each parameter class in the model, a representation of the derivation history is automatically induced using a form of multi-layered neural network. The parameters of the model are estimated using standard neural network methods for probability estimation, resulting in labelled precision/recall scores of 89.5%/88.8%.

## 2.2.7 Investigations into the Importance of Lexical Statistics

Although it is clear that modelling the dependencies between head words helps parsing, the exact contribution of lexical statistics was perhaps initially overestimated. Gildea [2001] describes experiments where, in a replication of Collins' Model 1, removing all bilexical statistics from the model<sup>1</sup> resulted in a mere 0.45% absolute reduction in  $f$ -score. Bikel [2004a] reports similar findings in his duplication of Gildea's experiments for his replication of Collins' Model 2. The work of Klein and Manning [2003] showed that an essentially unlexicalised history-based model could achieve accuracy rates as high as 86.3% for sentences  $\leq 40$  words on section 23. Their parsing model is based on a traditional PCFG but uses a Markov grammar and increases the amount of vertical conditioning context. They also add extra annotation of nodes, such as marking any nodes with unary productions with the suffix '-U', and they split some of the original Penn Treebank POS tags into several more fine-grained tags. In this latter step some POS tags actually come to represent a single word. However, the authors argue that this only occurs with functional word classes and so is not a lexicalisation of the model.

Bikel [2004a]'s investigation into the parameter classes of the Collins' model showed that, during parsing, for the  $P(w_i|H(i))$  parameter class the full context, that is the conditioning context which includes the head word of the phrase, was used only 1.49%

---

<sup>1</sup>This was done by removing the maximal context level in the interpolated estimates of  $P(w_i|H(i))$ , where  $w_i$  is the head word of the constituent  $i$  and  $H(i)$  its history.



	Authors	<= 40	<= 100	Description	Base Parser
Parsers	Collins [1999]	88.6%	88.2%	Generative history-based	
	Charniak [2000]	90.1%	89.5%	Generative history-based	
	Bod [2001]	90.7%	89.7%	DOP	
	Henderson [2003]	89.6%	89.1%	Generative history-based	
Rerankers	Collins [2000]	90.2%	89.7%	Boosting	Collins [1999]
	Collins and Duffy [2002]	89.2%	88.7%	Tree kernel	Collins [1999]
	Shen et al. [2003]	90.3%	89.8%	SVM LTAG	Collins [1999]
	Henderson [2004]		90.1%	Neural Networks	Henderson [2003]
	Charniak and Johnson [2005]		91.0%	MaxEnt	Charniak [2000]
	Koo and Collins [2005]		90.0%	Hidden Variable	Collins [1999]
	McClosky et al. [2006a]		92.1%	Self-Training	Charniak [2000]

Figure 2.1: Parser and Reranking F-score Results Comparison on Section 23 of the WSJ

of the time. The prevailing view at that point in time was that bilexical statistics are too sparse to make that much of difference to parsing accuracy. In further experiments Bikel found that although estimates were using bilexical statistics only 1.49% of the time, these statistics were being used up to 28.8% of the time during the generation of the top-scoring parse. The reason the bilexical statistics make such little difference to overall accuracy, Bikel argues, is that the distributions of the parameters which include the head word in the conditioning context and the parameters which omit that feature are so similar that it makes little difference which estimate is used. Although bilexical statistics may have limited usefulness, monolexical, or lexico-structural, dependencies, where syntactic structure is conditioned on the lexical head, appear to be more important with regard to parsing accuracy.

## 2.3 Ranking Algorithms

Two of the most accurate and popular state-of-the-art broad coverage statistical parsers are those of Collins [1999] and Charniak [2000]. They are very similar models - history-based generative parsers - and achieve F-scores of 88.2% and 89.5% respectively on the standard test set. Improvements in the accuracy of probabilistic parsers have occurred in very small increments over several years. However, since the publication of [Collins, 1999] and [Charniak, 2000] there have been no further improvements in accuracy reported for these parsers. Instead there has been a shift towards  $n$ -best parsing and discriminative reranking. In this approach the  $n$ -best list of parses for each



sentence output from a base parser are reranked using another, usually discriminative, model.

The usefulness of reranking the output of a base parser, with a model which can incorporate a richer feature set, was first demonstrated as early as [Ratnaparkhi et al., 1994]. Using a maximum entropy model, they reranked the trees output by the decision tree parser of Jelinek et al. [1994] and noted an improved score. Ratnaparkhi [1997] notes the potential of reranking by commenting on how the oracle score<sup>2</sup>, taken from as few as 20 top parses produced by a baseline parser, can be dramatically higher than the base parser's top-1 parse.

The recent shift towards discriminative reranking has been motivated to a large extent by the flexibility of discriminative reranking techniques in terms of feature selection compared to history-based models. Discriminative approaches can choose features which incorporate arbitrary aspects of the whole parse tree structure, whereas in history-based models the choice of conditioning features when predicting parse structure is limited to structure that has already been determined in the derivation of the tree.

Collins [2000] introduced a discriminative reranking approach for parsing: a ranking function is learned which assigns a ranking score to each candidate parse of a sentence from the  $n$ -best list of parses. Parse trees are represented by  $m$  binary valued features,  $h_k$ , for  $k = 1..m$ . The ranking function for a tree,  $x$  has the following form:

$$F(x, \bar{\alpha}) = \alpha_0 L(x) + \sum \alpha_k h_k(x) \quad (2.5)$$

where  $L(x)$  is the original log probability assigned the tree by the baseline parser.  $\bar{\alpha} = \{\alpha_0, \alpha_1, \dots, \alpha_m\}$  is a parameter vector of feature weights. The learning process involves finding the parameter weights that minimise some loss function, where the loss function is related to the number of ranking errors the ranking function makes on the training set. A ranking error rate is defined as the number of times a lower scoring parse (as measured against the gold standard parse) is incorrectly ranked above the best parse in the list. Experiments with two loss functions, one based on conditional Markov fields and another based on the boosting algorithm, are made in [Collins,

---

<sup>2</sup>The oracle score is the score that an 'oracle' would get were it to pick from each  $n$ -best list the highest scoring parse according to measures of labelled precision/recall. The oracle score is less than 100% because the correct parse is not always among the top- $n$  parses produced by a parser. The oracle score marks the upper accuracy limit for rerankers.

2000] and presented in more detail in [Collins and Koo, 2005]. The baseline parser used to produce the  $n$ -best lists is that of Collins [1999]. A rich feature set of over 500,000 binary features was used in the final model. The boosting algorithm approach increases precision/recall accuracy from 88.3%/88.1% labelled precision and recall to 89.9%/89.6%.

The discriminative reranker of Charniak and Johnson [2005] follows the reranking methodology of Collins [2000]. Feature weights are trained using a maximum entropy estimator. The parser of Charniak [2000] is adjusted to output the 50-best parses (see §2.3.1) for a sentence. A set of carefully hand-crafted features types are designed and include features which, for example, capture a preference for parallelism across conjuncts (see §2.6) and right-branching trees. In [McClosky et al., 2006a] the accuracy of this reranker is further improved to an impressive 92.1%  $f$ -score through a self-training method. The reranker of Charniak and Johnson [2005] was used to parse sentences from an unannotated corpus of a similar newswire domain. The 1-best parse trees produced by the reranker were then added to the original WSJ hand-parsed corpus, and the resulting enlarged corpus used to re-train the original base parser of Charniak [2000]. In this process events from WSJ trees were given more weight than events from reranker-produced trees. Re-training the parser in this fashion increased the accuracy of the base parser to 91.0%. Finally, a reranker which used this higher-accuracy parser as a base parser achieved the highest accuracy to date on the WSJ test set.

Reranking tree kernel approaches have also been developed such as that of Collins and Duffy [2002], where features consist of all possible subtrees, as in [Bod, 1998], and the voted perceptron algorithm is used to learn the feature weights. Another successful tree kernel approach is that of Shen et al. [2003] who use support vector machines and Lexicalized Tree Adjoining Grammar based features.

In an effort both to reduce data sparsity and to handle polysemous words, Koo and Collins [2005] propose a conditional log-linear model with hidden variables which represent the assignment of words to word clusters or word senses. The input to this model is the  $n$ -best trees produced by Collins [1999]’s parser. When combined with the base parser Collins [1999] and features from the Collins [2000] reranker the log linear model gives a small improvement over Collins [2000].

In Henderson [2004] the accuracy of a neural network generative parser [Henderson,

2003] is improved on when its top 20 parses are reranked by a discriminatively trained model. Instead of training the parameters of the generative model by maximising the joint likelihood of the trees and the sentences of the training corpus, the parameters of the joint model are trained by maximising the conditional likelihood of the parses in the corpus given the sentences in the corpus, resulting in improved performance.

Advantages of ranking algorithms as opposed to generative probabilistic approaches are that rerankers are relatively straightforward to implement and, importantly, that it is trivial to incorporate arbitrary features in a ranking setting whereas adding new features to a generative parser can be difficult.

### 2.3.1 $n$ -best lists

Generally, the better the  $n$ -best list of candidate parses produced by a base parser the better a reranker will do, where the quality of a list can be measured by its oracle score. A higher oracle score tends to be correlated with a higher reranker score. The  $n$ -best lists for the experiments in [Collins, 2000] were produced by simply turning off dynamic programming in the chart parser of [Collins, 1999] (by not allowing any two edges to be equal). This, of course, slows down parsing prohibitively and so the beam width<sup>3</sup> was narrowed from  $10^{-4}$  to  $10^{-3}$  and a chart cell limit of 100 was imposed. In addition to the  $n$ -best lists produced in this fashion, the 1-best output produced by the original Collins [1999] parser was added to the mix. However, as demonstrated in [Huang and Chiang, 2005], restricting the search space in such a fashion affects the quality of the  $n$ -best lists produced. Huang and Chiang [2005] develop new efficient algorithms for producing high-quality  $n$ -best lists. Building on top of Bikel's implementation of Collins' parser they could produce 10000-best lists in almost the same time as 1-best lists and in experiments with  $n = 100$  and a beam width of  $10^{-4}$  achieved an oracle  $f$ -score of 96.4%, compared to Collins' 94.9%. This algorithm was adopted for the reranking of McClosky et al. [2006a] over the original method used for the Charniak and Johnson [2005] reranker which had simply kept the  $n$ -best edges, rather than the 1-best, during the second-pass of the parser.

---

<sup>3</sup>In [Collins, 1999] in order to increase efficiency, a beam width is used to prune the search space. Any constituent whose probability is less than  $1/10000$  times the highest probability constituent for the same word span is pruned from the search space.



## 2.4 Memory-based Learning and Natural Language Processing

We now look at the learning technique adopted in this thesis - memory-based learning. Learning techniques that come under the relatively broad category of memory-based learning have been applied to a variety of language learning tasks. For example, nearest neighbour techniques have been applied with some success to PP-attachment [Zavrel et al., 1997, Zhao and Lin, 2004] and shallow parsing [Daelemans et al., 1999b, Sang, 2002]. In this section we begin by outlining the argument that memory-based learning is particularly suited to natural language learning tasks. We then give a brief summary of memory-based techniques applied to full parsing. The next section on similarity for smoothing (Section 2.5) includes also previous work on  $k$ -NN for smoothing.

### 2.4.1 Advantages of Local Learning for Natural Language Learning Tasks

There is evidence that in many natural language learning tasks the instance space tends to be highly disjunctive [van den Bosch et al., 1997, Daelemans et al., 1999a, Daelemans and van den Bosch, 2005]. That is, natural language data sets often contain many small disjuncts, where a disjunct is a cluster of identically classified instances. In a highly disjunctive instance space those disjuncts that correctly classify only a few training examples collectively cover a significant portion of the text. Daelemans et al. [1999a], for example, measure the degree of disjunctivity of several data sets (grapheme-to-phoneme conversion, part-of-speech tagging, PP-attachment, and base noun phrase chunking) in the following leave-one-out experiments: For each instance in a data set, the 50 nearest neighbours to the instance are retrieved from the remaining data set and ranked according to their distance to the left-out instance. The cluster size of the left-out instance is measured as the rank (minus one) of the nearest neighbour to the left-out instance that has a different class value to the left-out instance. Using this method to measure disjunctivity, many different NLP data sets were shown to be highly disjunctive.

Studies such as [Weiss, 2000] have shown that small disjuncts are much more error prone than large disjuncts and contribute to a disproportionate number of the total

errors. Aha [1992]'s analysis shows how the  $k$ -nearest neighbour algorithm performs well at highly disjunctive learning. The work of [Daelemans et al., 1999a, Daelemans and van den Bosch, 2005] shows that variants of  $k$ -NN work well on several different natural language learning tasks and relate this to the fact that lazy learning retains all information concerning disjuncts, no matter how small, whereas eager-learning algorithms such as decision trees implement pruning and tend to overgeneralise thus losing important disambiguating information contained in small disjuncts. Thus nearest neighbour, because it learns locally, would seem to be an ideal candidate for natural language learning.

The advantages of local learning for natural language learning are demonstrated in Daelemans et al. [1999a] in a series of experiments which showed that editing exceptional instances from the training set tended to harm generalisation accuracy, although similar experiments [Rotaru and Litman, 2003] on learning tasks in the area of spoken dialog systems did not show such clear evidence for the harmful effect of editing exceptional instances (but see §2.5.2 for evidence of the importance of rare events in other types of similarity-based learning). In another series of experiments in [Daelemans et al., 1999a],  $k$ -NN learning was shown to outperform decision tree learning, and decreases in the performance of the decision tree classifier were shown to be linked to the degree of abstraction from exceptions (by pruning or the eagerness of the algorithm).

## 2.4.2 Memory-Based Parsing

Scha et al. [1999] show how Data-Oriented-Parsing (DOP) [Bod, 1998] relates to memory-based learning techniques such as Case-Based Reasoning (CBR), noting that although DOP differs from mainstream CBR methods, there are some similarities: DOP is lazy as it does not generalise over the treebank until it starts parsing a new sentence, and DOP defines a space of parses for a new input sentences simply by matching and combining fragments from the treebank.

DOP models, instead of defining their probabilities on minimal syntactic rules as with traditional probabilistic grammars, define probabilities over whole trees, and tree fragments. The main motivation for this is the belief that syntactic constructions of arbitrary size and complexity may be statistically important. DOP is not an extension of PCFGs but is based on a grammar formalism known as Tree Substitution Grammar. In

theory, in the DOP framework all previously seen parse trees and parse tree fragments can be used to construct and assign probabilities to a new parse tree. In practice, the space of tree fragments used for training is restricted somewhat as this tends to increase accuracy. Bod [2001] notes that whereas the parsing approaches of Collins [1999] and Charniak [2000] limit the statistical dependencies beforehand (in that a limited set of conditioning features are hard-coded in advance) and then extend the dependencies in order to increase accuracy, DOP instead begins by taking into account all fragments seen in the treebank and then experiments with restrictions in order to uncover the optimal set of relevant fragments. As displayed in Table 2.1 a version of DOP [Bod, 2001] tested on the Penn treebank recorded high accuracy results.

Sang [2002] achieved state-of-the-art results for base noun phrase identification, arbitrary base phrase recognition, and clause detection using combinations of  $k$ -NN classifiers. However, when the cascaded memory-based approach (where the output of one level of chunking is used as the input to the next level) was applied to full parsing the results were below state-of-the-art parsers, receiving an  $f$ -score of 80.5% on section 23 of the Penn WSJ treebank.

Kübler [2004] describes a parser based on memory-based learning, trained and tested on the TüBaD treebank [Stegmann et al., 2000, Hinrichs et al., 2000], a treebank made up of speech transcripts in several domains. A test sentence is first POS tagged and divided into syntactic chunks using the chunk parser of Abney [1996]. The memory-based parser then searches for the most similar sentence in the training set based on sequences of words. Training instances are stored in a prefix trie of words and the search for the most similar sentence is a search for the most similar sequence of words in the trie. If no reasonably similar sentence is retrieved then a backoff module searches for similarity based on chunk information. When searching for the most similar sentence in the training set the search algorithm allows ignoring words or chunks in both the new sentence and a training sentence when the exact match cannot be found. The retrieved sentence, with its associated parse tree, is then adapted to the test sentence, by omitting the words or phrases that were omitted in the search. The parser gets a labelled precision/recall  $f$ -score of 84.78%. The data set for the experiments contains sentences of average length 4.5 words which is very low compared to the average length of WSJ treebank sentences (23 words) and, as discussed in [Kübler, 2006], the parser, as is, would not be suitable for unrestricted newswire text because



of the extreme unlikelihood of finding the same or very similar sentence to the test sentence in the training set. Dialog data, on the other hand, contains many repetitions and has fewer words per sentence.

## 2.5 Similarity for Smoothing

We now turn to the area of probability estimation using similarity for smoothing. We first look at previous work on  $k$ -NN for smoothing and then to the related area of nearest neighbour cooccurrence smoothing.

### 2.5.1 $k$ -NN for Smoothing

Although they do not apply  $k$ -NN to probabilistic estimation, Zavrel and Daelemans [1997] explore the relationship between  $k$ -NN and non-interpolated back-off smoothing. They demonstrate that when  $k=1$  and an unweighted overlap metric used, then  $k$ -NN and the backoff model both specify similar hierarchies of abstraction of the context features; that is, the ordering of the feature subsets in back-off smoothing is identical to the ordering of buckets of neighbours in the  $k$ -NN algorithm. Zavrel and Daelemans argue that memory-based learning has advantages over backoff smoothing in that the similarity metric and feature weighting scheme automatically specify a domain-specific hierarchy between the most specific and most general conditioning information and do so without the need for a large number of parameters.

Toutanova et al. [2003] report that nearest neighbour techniques outperformed decision trees and other smoothing methods (Witten-Bell, Jelinek-Mercer, and log-linear models) when used to estimate local probability distributions in history-based generative parsing models. As the Redwoods HPSG treebank was used, results are not directly comparable with parsers based on the Penn Treebank. Nonetheless, results are relevant to Penn Treebank parsing because of the use of similar generative models, which have some conditioning features in common with typical Penn Treebank parsers.

Toutanova et al. [2003] also make explicit the relationship between deleted interpolation models and a broad class of memory-based learning. Their analysis is based on a nearest neighbour model which is restricted to a linear order among feature subsets, and which uses the overlap distance function to measure distance between instances.

They show that memory based models of this type are a subclass of deleted interpolation models, where the value of the backoff weights in the interpolated models is based on counts of feature subsets and distance-weighting. In practice interpolated models use strictly linear feature subsets for the various levels of conditioning feature backoff. Similarly restricting the features in the  $k$ -NN algorithm allowed a direct comparison of results. The  $k$ -NN model worked best at  $k=15,000$ , outperforming the other smoothing techniques. Accuracy was again increased when the features were not restricted to a linear order.

## 2.5.2 Cooccurrence Smoothing

The problem of estimating the probability of events from sparse data is particularly severe when the events involve individual words. It comes as no surprise then that in the area of language modelling for speech recognition, where it is necessary to estimate the probability of a sequence of words, the smoothing of estimates has long been a major focus. Several smoothing techniques have been developed (see for example [Jelinek and Mercer, 1980, Katz, 1987, Church and Gale, 1991]), including class-based approaches such as that of [Brown et al., 1990], where words are clustered into classes and the probability of a cooccurrence is determined using the probability of class cooccurrences. In this subsection we focus on cooccurrence smoothing because of its similarity with one of the estimation techniques developed in this thesis (§4.3.2). We discuss how our word similarity-based smoothing relates to cooccurrence smoothing in (§4.4).

Cooccurrence smoothing for language modelling was first introduced by Essen and Steinbiss [1992]. The basic idea is that when estimating the conditional probability of word bigram,  $P(w_2|w_1)$ , cooccurrences of word  $w_2$  with words similar to  $w_1$  can be useful. The cooccurrence-smoothed estimate takes the form:

$$P_{co-smooth}(w_2|w_1) = \sum_{w'_1} P(w_2|w'_1)f(w_1, w'_1) \quad (2.6)$$

where  $f(w_1, w'_1)$  depends on the similarity of  $w_1$  and  $w'_1$  and, in this case, is the confusion probability. In [Essen and Steinbiss, 1992] the estimate in (2.6) is combined with maximum likelihood estimates by way of linear interpolation to give the final smoothed estimate. This technique was used in [Grishman and Sterling, 1993] for smoothing

models of selectional constraints; that is, smoothing probabilities of words occurring together in specific syntactic relations.

In [Dagan et al., 1994] cooccurrence smoothing for language modelling is further explored. Their similarity model has the following form:

$$P_{SIM}(w_2|w_1) = \sum_{w'_1 \in S(w_1)} P(w_2|w'_1) \frac{sim(w_1, w'_1)}{\sum_{w'_1 \in S(w_1)} sim(w_1, w'_1)} \quad (2.7)$$

where  $S(w_1)$  is the set of nearest neighbours of  $w_1$  and  $sim(w_1, w'_1)$  is a similarity function derived from the *Kullback-Leiber* (KL) distance between the probability distributions of  $w_1$  and  $w'_1$ . This method of similarity-based smoothing is also called *distance-weighted averaging* [Lee, 1999]. The smoothed estimate,  $P_{SIM}(w_2|w_1)$ , is a combination of estimates for cooccurrences involving words similar to  $w_1$ , where each estimate is weighted by a normalised measure of similarity between  $w_1$  and neighbour  $w'_1$ . The nearest neighbour set  $S(w_1)$  was chosen to be the set of, at most,  $k$  words that were less than a certain distance  $t$  from  $w_1$ .  $k$  and  $t$  were tuned experimentally. Dagan et al. [1994] found the best results when combining the estimate in (2.7) with the unigram probability  $P(w_2)$  via linear interpolation. This interpolated estimate was used only for the estimation of the probability of bigrams that had never before occurred in the training data. Otherwise, in cases where a bigram had occurred before in the training set, the maximum likelihood estimate,  $P_{MLE}(w_2|w_1)$ , was used. These two estimators were combined using a variation of Katz [1987]'s back-off model. The estimation technique was found to be effective in the task of language modelling leading to a reduction in perplexity and speech-recognition error.

In an continuation of this work [Lee, 1997, Dagan et al., 1999], experiments are carried out with an additional three similarity functions (the confusion probability, the L1 norm and the Jensen-Shannon divergence) and their success evaluated on a pseudo word disambiguation task. The performance of the similarity-based methods were found to be on the whole better than that of standard methods. Interestingly, they found that when events which occurred only once in the data were omitted from the training set, the similarity-based smoothing methods suffered noticeable performance degradations. As noted in §2.4.1, a similar phenomenon - that rare events are useful - was found in case editing experiments with the  $k$ -NN algorithm [Daelemans et al.,



1999a].

Lee [1999] compares seven distributional similarity measures in a restricted version of the distance-weighted averaging model to see which measure is best at returning useful nearest neighbours. Based on an analysis of the results a new similarity function, the  $\alpha$ -skew divergence, is developed which performs better than the other functions.

More recent work on cooccurrence smoothing [Weeds, 2003a] (and in more detail [Weeds, 2003b]) has experimented with how to choose the set of nearest neighbours  $S(w_1)$  in (2.7). The estimates  $P(w_2|w'_1)$  were combined as in (2.7) as each neighbour  $w'_1$  was added to  $S(w_1)$  until some stopping condition was met. Experiments with three different stopping conditions were made and results showed all techniques gave an improvement over maximum likelihood estimation and naïve Add-one type smoothing.

## 2.6 Previous Work on Coordination Ambiguity Resolution

We now conclude this chapter with a review of previous work on coordination disambiguation.

Agarwal and Boggess [1992] present a deterministic algorithm for identifying the conjuncts of coordinating conjunctions. Their conjunct identifier is a component of an expert system for the automatic extraction of information from structured reference manuals, in this case the Merck Veterinary Manual. The task is to identify conjuncts which appear in text that has been part-of-speech tagged, as well as tagged with semantic labels specific to the domain. The text has also been processed by a semi-parser which identifies noun, verb, prepositional, gerund, adjective, and infinitive phrases in the sentences. The algorithm makes the simplifying assumption that each coordinating conjunction conjoins only two conjuncts. The post-CC conjunct is always taken to be the first complete phrase that follows the CC. The identification of the pre-CC conjunct is based on heuristics which essentially work from the CC backwards to the start of the sentence and choose as the first conjunct the first word or phrasal component which matches the second conjunct's semantic and syntactic labels, relaxing the constraints to match syntactic labels only if no such component is found. As the semi-parser does not recognise clauses and some phrases the conjunct identifier was expected only

to correctly recognise the beginnings of clauses and phrases being conjoined but not the right boundary of the components. Within this framework the system correctly identified 81.6% of coordinations on a test set of some 544 cases.

In [Kurohashi and Nagao, 1994] coordination disambiguation is carried out as the first component of a Japanese dependency parser using a technique which calculates similarity between series of words from the left and right of a conjunction. Similarity is measured based on matching POS tags, matching words and a thesaurus-based measure of semantic similarity. The most similar two series of words is calculated using a dynamic programming technique. Their method first identifies the coordinate structures in a sentence and then performs a dependency analysis for each phrase in the identified structures. Each conjunctive structure is then reduced to a single node and a deterministic dependency analysis is carried out on the reduced sentence.

Resnik [1999] considers noun phrase coordinations of the form  $n1$  and  $n2$   $n3$  and of the form  $n1$   $n2$  and  $n3$   $n4$ . The former has two possible structural analyses, for example *a (bank and warehouse) guard* and *a (policeman) and (park guard)* while for the latter, five alternate structural analyses are possible. The learning task was to determine which nouns are the heads being coordinated. In the first example given above,  $n1$ :bank and  $n2$ :warehouse are coordinated. In the second example  $n1$ :policeman and  $n3$ :guard are the conjoined heads. In the Penn Treebank, following the guidelines [Bies et al., 1995], both types of noun phrase would be given a flat structure. Thus for the validation and test sets a selection of noun phrases of these forms were extracted from the Wall Street Journal corpus and disambiguated by hand. There were 100 and 89 samples in the two test sets respectively. To resolve the ambiguities automatically Resnik uses three main sources of information: agreement in number of candidate conjoined nouns, similarity of meaning between the nouns, and a measure of the appropriateness of noun-noun modification. Experiments were carried out on various different ways of combining these sources of information, including using a heuristic method as well as a decision tree approach. For the noun phrases involving three nouns 80% of the coordinations were disambiguated successfully with 100% coverage. For the second type of coordinate noun phrase accuracy of 81.6% was achieved with 85.4% coverage.

In Goldberg [1999] an unsupervised statistical model for disambiguating coordinate noun phrases of the form  $n1$   $p$   $n2$   $cc$   $n3$  is presented. Here the problem is framed as an attachment decision: does  $n3$  attach 'high' to the first noun,  $n1$ , or 'low' to  $n2$ ? In the



noun phrase *box of chocolates and roses* ‘roses’ attaches high to ‘box’ yielding: ((box (of chocolates)) and roses)). In *busloads of executives and their wives* ‘wives’ attaches low to ‘executives’ giving the structure (*busloads (of ((executives) and (their wives)))*). A maximum entropy statistical model based on Ratnaparkhi [1998b] is used to estimate the probability:  $Pr(a|n1, p, n2, cc, n3)$ , where the variable  $a$  is either a high attachment or low attachment. The model is trained from examples of unambiguous coordination noun phrases, of the form  $n1\ cc\ n2$ , extracted from the unannotated WSJ. The results presented had not yet been tested on a separate test set, but achieved precision of 72% on the 500-phrase validation set extracted from the WSJ Treebank.

Nakov and Hearst [2005] focus on disambiguating noun compound coordination of the form  $n1\ cc\ n2\ n3$ , where the CC is limited to *and* or *or*. They consider phrases such as *car and truck production* where both *car* and *truck* modify the head noun *production*. These phrases are given a flat structure: (car and truck production). Alternatively, for phrases such as *president and chief executive*, a two-headed noun phrase, they assign the following bracketing: ((president) and (chief executive)). The task is to decide which of the two possibilities is the correct bracketing structure given the four words. In the case where the nouns in a noun coordinate construction are modified by non-nominal modifiers, these modifiers are used for disambiguation but the final bracketing deals only with the four words outlined above. The Web is used to get statistics on word cooccurrences such as counts on how often  $n1$  occurs with  $n3$  compared with  $n2$  cooccurring with  $n3$ . Counts from the Web of paraphrase patterns are also used as an information source. For example if the pattern  $n2\ cc\ n1\ n3$ , as in *truck and car production*, is matched often enough then a flat structure is likely. The Web based statistics are used to make decisions on which type of bracketing should be employed. In addition, heuristics which take into account any adjectives or determiners modifying the nouns as well as a number agreement heuristic all have a vote on the correct bracketing. The final decision is based on a majority vote from all information sources, with a default of flat structure when the various sources of information were undecided as to the correct bracketing structure.

The system was tested on 428 examples apparently automatically extracted from the Penn Treebank and achieves 80.6% accuracy. Officially, however the Penn Treebank would assign a flat structure to both examples given above because they only

involve nominal modifiers<sup>4</sup>, although in practice this is often not the case (see discussion in §6.2). If a non-nominal unshared modifier is introduced, as in *president and old chief executive* then the phrase is given added structure: ((president) and (old chief executive))<sup>5</sup>. The results of Nakov and Hearst might be obscured somewhat by not accounting for flat structure often given to both types of noun coordination in the Penn Treebank. Only when the noun phrases in their test set originally contained unshared non-nominal modifiers should the treebank give the more structured bracketing.

Interestingly, the early discriminative parse reranker of Ratnaparkhi et al. [1994] includes features to capture parallelism in conjuncts. The model has classes of boolean features which return 1 if particular syntactic patterns occur in both conjuncts. The discriminative reranker of Charniak and Johnson [2005] also included features to capture syntactic parallelism across conjuncts at various depths. One type of feature indicated whether conjuncts had the same syntactic label at depth 0, whether they had the same labels for all nodes at both level 0 and level 1 and so on. Another feature type measured the difference in the number of preterminals dominated by two conjuncts. An extra boolean flag indicated whether the two conjuncts in question were the last two in the coordinated phrase. For both rerankers, results were not given for the effect of the coordination-based features alone, but rather for the effect that these and other features had on overall reranker scores.

---

<sup>4</sup>See, for example, page 138 of the Penn Bracketing Guidelines [Bies et al., 1995]. ‘In general, we avoid showing either the internal structure or the extent of modification of noun modifiers, regardless of the strength of the annotator’s intuition in a particular example...[In the case of an NP with multiple heads] if the only unshared modifiers are nominal, we annotate with flat structure’.

<sup>5</sup>See page 139 of the Penn Bracketing Guidelines [Bies et al., 1995]. ‘When there are unshared modifiers, added structure shows which modifiers go with which head...When there are unshared adjectives, determiners, or possessives, we frequently end up showing structure for nominal modifiers as well’.

## Chapter 3

# Memory-Based Parameter Estimation

### 3.1 Introduction

In this chapter we describe a history-based generative parsing model which uses a  $k$ -nearest neighbour technique to estimate the model's parameters. Taking the output of a base  $n$ -best parser we use our model to re-estimate the log probability of each parse tree in the  $n$ -best list for sentences from the Penn Wall Street Journal treebank. By further decomposing the local probability distributions of the base model, enriching the set of conditioning features used to estimate the model's parameters, and using  $k$ -NN for estimation as opposed to the Witten-Bell estimation of the base model, we achieve an  $f$ -score of 89.4%, representing a 6% relative decrease in  $f$ -score error over the 1-best output of the base parser.

### 3.2 Motivation

As discussed in §2.4, previous work has shown memory-based learning to be effective for natural language learning tasks. Of particular relevance to the work presented here is the work of Zavrel and Daelemans [1997] and of Toutanova et al. [2003]. Zavrel and Daelemans [1997] discuss the advantages of  $k$ -nearest neighbour for smoothing over linear interpolation, and show how  $k$ -NN gives a convenient way of experimenting with

complex conditioning events. In the work of Toutanova et al. [2003] on parameter estimation in a Head-Driven Phrase Structure Grammar (HPSG) parsing model [Polar and Sag, 1994], estimation using  $k$ -NN outperformed other estimation techniques (Witten-Bell, Jelinek-Mercer, decision trees and log-linear models).

Given the stated advantages of  $k$ -NN over linear interpolation and motivated by the success of  $k$ -NN for estimation in the HPSG parsing domain, we applied  $k$ -NN to the task of smoothing local probability distributions in a generative PCFG-derived history-based parsing model. For a baseline model, we replicated the state-of-the-art generative parsing model of Collins [1999] (hereafter the Collins parser). This high-accuracy parser uses Witten-Bell linear interpolation for parameter estimation. Though the model is history-based there are relatively few conditioning features in the parameter classes of the model, presumably due to the limitations of linear interpolation for smoothing. Estimating the parameter classes with  $k$ -NN could potentially improve the model by allowing for better smoothing of data and for more information from the history to be used in predictions.

### 3.3 The Baseline Model

We replicated Model 1 of the Collins parser so that we would then have a baseline against which we could test the effect of any subsequent alterations we made to the model. Although Model 2 of [Collins, 1999] achieves more accurate results than Model 1, we chose to replicate Model 1 as our baseline model. Model 2 of [Collins, 1999] introduces probabilities over subcategorisation frames in order to solve the problems caused by the bad independence assumption in Model 1: that modifiers are generated independently of each other. However, given that  $k$ -NN estimation should allow for greater freedom with conditioning features we anticipated increasing the conditioning context for the generation of modifiers so that the history would include previously generated sibling modifiers (i.e. by increasing the level of horizontal markovisation). This could weaken the independence assumptions of Model 1 and yet avoid the added complication of modelling subcat frames. Our replication of Model 1 is our first baseline against which we compare the memory-based model. In our evaluation of results we



also compare the memory-based model with Collins Model 2.<sup>1</sup>

Replicating Model 1 of the Collins parser proved to be a more complicated task than originally anticipated. However, based on the details published in [Collins, 1999], as well as those published in [Bikel, 2004b], we replicated parsing Model 1. We reproduced all the preprocessing steps on the trees in the training and test sets necessary for the model and then, given a preprocessed parse tree, our parsing model will estimate the probability of the tree, following the parameterisation of the Collins parser.

As our research is concerned with the generative parsing model and the estimation of its parameters, rather than the mechanisms of the parser itself, and given the availability of Bikel's replication [Bikel, 2002] of the Collins Model 2 parser with its  $n$ -best list producing option, it was not necessary to reproduce the actual parser.<sup>2</sup> Instead, we rerank the output of Bikel's  $n$ -best parser, first according to our replication of Collins Model 1 and then according to our memory-based model. As Bikel's parser is based on Model 2, we would not expect our Witten-Bell Model 1 replication to give a better ranking of parses. This step was necessary however in the development of the parsing model in order to give us our own baseline parsing model mechanism.

### 3.4 The Memory-Based Model

The memory-based parsing model differs from the Collins parser (Model 1) in how parameters are estimated: instead of Witten-Bell estimation, we use  $k$ -NN for estimation, as described in §1.4.2. We return to the details of our estimation technique in the next section. The memory-based model also differs from Model 1 in that it is parameterised slightly differently, with increased conditioning features. History-based models allow for conditioning on any previously built structure, therefore vertical markovisation, where information from previously generated ancestor nodes is used as part of the local history in a parameter class, can be employed to improve the predictive capacity of a parameter class. In our model the feature sets for each parameter class are expanded to include additional features from the parse history, specifically increasing

---

<sup>1</sup>Collins [1999] also presents a third parsing model, which models traces and Wh-movement, however it does not improve on the accuracy of Model 2.

<sup>2</sup>Initially, however, in a proof of concept experiment, we replicated the actual parser for Model 1, achieving similar results to Collins Model 1 results. This allowed us to test that we had correctly replicated Model 1, though in subsequent experiments we did not use the actual parsing mechanism.



the  $n^{\text{th}}$ -order assumption for both horizontal and vertical markovisation. Choosing  $n$  greater than 1 in vertical markovisation has been shown to be useful in the parsing models of [Johnson, 1998, Charniak, 2000, Klein and Manning, 2003]. In the latter two works using a  $3^{\text{rd}}$  and (variable)  $2^{\text{nd}}$  order horizontal markovisation, respectively, proved optimal. Another difference between our model and the Collins parser Model 1 is that where Collins estimates the constituent label of a dependent constituent, its part-of-speech tag, and its punctuation and coordination flag in one step, we do so in three steps: first the probability of the part-of-speech tag is estimated, then the constituent label, and finally its coordination and punctuation flags. Details of the parameter classes of the memory-based model are given in Section 3.5. See [Bikel, 2004b] for a detailed description of all the parameter classes in the Collins parser.

### 3.4.1 Constraint Features for Training Set Restriction

The number of training examples in the training set for a particular parameter class can be quite high. For example, the number of head child generation events in the parse trees of sections 02 to 21 inclusive of the Penn WSJ treebank is 947,715 and the number of examples of the generation of a constituent to the right of a head child is 1,367,411. In  $k$ -NN estimation each time a conditional probability is estimated it is necessary to calculate  $\Delta(X, X')$ , that is the number of feature value mismatches between the history of the probability we wish to estimate and the history of an instance from the training set, for each distinct example in the training set for this parameter class.

We found that restricting the number of examples in the training set used in a particular parameter estimation helped both in terms of accuracy and speed. We restricted the training sets by making use of constraint features. Take, for example, the head child generation parameter class which estimates the probability of head child label  $C_h$  given, say, a history  $C_p, w_p, t_p, t_{gp}$ , where  $C_p$  is the parent non-terminal label,  $w_p$  and  $t_p$  its head word and part-of-speech respectively, and  $t_{gp}$  is the POS tag of the grandparent node. If we make  $C_p$  a constraint feature then to estimate, say,  $P(C_h = IN | C_p = PP, w_p = at, t_p = IN, t_{gp} = NN)$  we would make our prediction using only those training examples where  $C_p = PP$ . This mechanism is somewhat similar to the idea of the MAC/FAC model of similarity-based retrieval [Gentner and Forbus, 1991], which is based on a two-stage process, where a computationally cheap

filter is used to retrieve a subset of similar examples which will undergo more expensive processing (similar also to the idea behind multiple-pass parsing [Goodman, 1997, Charniak, 2000]). Stanfill and Walz [1986] make use of constraint features which they call predictor restriction in order to increase the accuracy of their memory-based reasoning approach to word pronunciation. They use a feature weighting function to determine the most important feature which they then use to restrict their training set to only those examples which have the same value for that feature as the query instance. Daelemans et al. [1997] also explore a constraint feature approach to  $k$ -NN. Their TRIBL algorithm constrains the initial data set based on features chosen according to their information gain and then classifies a query instance based on applying  $k$ -NN to the reduced training set. TRIBL performed well on several non-linguistic data sets achieving similar accuracy results to  $k$ -NN without data set restriction but with considerable speed advantages over the standard  $k$ -NN algorithm. Our constraint features were generally chosen to coincide with the features in the last level of back-off of the parameter classes in the Collins parser.

We carried out experiments using different sets of constraint features, some more restrictive than others. The mechanism we used is as follows: if the number of examples in the training set, retrieved using a particular set of constraint features, exceeds a certain threshold value then use a higher level of restriction i.e. one which uses more constraint features. If, using the higher level of restriction, the number of samples in the training set falls below a minimum threshold value then ‘back-off’ to the less restricted set of training samples.

### 3.4.2 Smoothing

Although using  $k$ -NN for parameter estimation is an effective way of smoothing a probability estimate with many conditioning events, it is nevertheless still necessary to smooth the  $k$ -NN estimate. This is in order to avoid a zero probability when the class value of the query case does not occur in any of the events selected for the estimation of the parameter. Following Toutanova et al. [2003], in order to avoid zero probabilities we added artificial instances to the training set, one for each class value. These instances are made to be at a certain distance from the query instance, a distance considerably larger than the maximum distance of a ‘real’ training instance from the query instance.

### 3.4.3 Lexical Statistics

The word parameter class,  $P(w_i|H(i))$ , where  $w_i$  is the head word of the node being generated and  $H(i)$  is its history, is distinguished from all others in the relatively large number of class values possible (12,002 possible values). We found that the estimation of this parameter class required a slightly different treatment to that of the other parameter classes. The best results were obtained when the  $k$ -NN estimate was combined with the original Witten-Bell estimate of Model 1. Let  $X_1$  be a feature vector which is a particular instantiation of the history context of the parameter class  $P(w_i|H(i))$ .  $P(w_i|X_1)$  is calculated as follows:

$$P_{interp}(w_i|X_1) = \lambda_{Y_1} P_{kNN}(w_i|X_1) + (1 - \lambda_{Y_1}) P_{WB}(w_i|X_2) \quad (3.1)$$

where  $Y_1$  is a constraint feature vector, as described in §3.4.1. The constraint features in  $Y_1$  are a subset of the features in  $X_1$ .  $X_2$  is a feature vector, less specific than  $X_1$ , to which we back off. The calculation of the  $\lambda$  weights is similar to the weight calculations in the Witten-Bell interpolation, outlined in §1.4.1. The interpolation weights here are derived from the count of the constraint feature values in the training set. We define  $\lambda_{Y_i}$  in terms of  $|Y_i|$ , which is the number of times context  $Y_i$  occurs in the corpus:

$$\lambda_{Y_i} = \begin{cases} 0 & \text{if } count(Y_i) = 0 \\ \frac{count(Y_i)}{count(Y_i) + C * D(Y_i)} & \text{if } count(Y_i) > 0 \end{cases}$$

where  $C$  is a constant which can be optimised using held-out data.  $D(Y_i)$  is the diversity of the history  $Y_i$ , that is the number of distinct outcomes that have been seen with context  $Y_i$  in the training sample. In practise,  $\lambda_{Y_i} = 0$  does not occur. The constraint feature for this parameter class is chosen to be  $t_i$ , the POS tag of  $w_i$ . In the training set there is no POS tag value for which we have no events. As in this case the  $k$ -NN estimate is smoothed by backing off to the Witten-Bell estimate it is not necessary here to add the artificial instances to the training set, as described above in Section 3.4.2.



## 3.5 Experiments

### 3.5.1 Experimental Set up

Following established methodology our model is trained on sections 02 to 21 inclusive of the Penn WSJ treebank and tested on section 23. Initially we used section 24 as the validation set. However we encountered overfitting problems with this data set and subsequently switched to using sections 00, 01, 22 and 24 for validation. In order to speed up the fine-tuning process we chose the set of top- $n$  parses for every third sentence in this set so that the final validation set contained sentences from each of the four sections. As in this phase of experiments we test on sentences of length less than or equal to 40 words, the validation set is also made up only of parses for sentences less than 41 words. The validation set was used for development and parameter tuning.

For the validation and test sets we obtained a set of top- $n$  parses from the Bikel parser setting the beam width to  $10^{-3.5}$ , getting an average of 31 parses per sentence. We then merged this with the 1-best output of the same parser run with a wider beam of  $10^{-4}$ . According to Huang and Chiang [2005], Collins [2000] uses a similar process for producing such a merged  $k$ -best list. This gave us an oracle  $f$ -score of 96% for section 23, on sentences  $\leq 40$  words.

As discussed further in §2.3.1, Bikel's implementation of the Collins parser includes a 'hack' (which essentially turns off dynamic programming) to allow for  $n$ -best parsing. As in [Huang and Chiang, 2005] we found that with the standard beam width of  $10^{-4}$  the method was prohibitively expensive. We limited our experiments to sentences of  $\leq 40$  words because we found that for longer sentences we could not produce good quality  $n$ -best lists. If we narrowed the beam width too much the oracle score for the parses produced was not high - that is, the quality of the  $n$ -best lists was poor. Yet, if we widened the beam width we either encountered memory problems or the time taken to produce the list was prohibitively long. (See §3.7 for the specifications of the machine we used).

### 3.5.2 Experimental Details

We re-estimated the probability of each parse using our own baseline model, which is a replication of the Collins parser Model 1, with the same parameter classes, and

using the same Witten-Bell estimation technique. We tested  $k$ -NN estimation first on the head generation parameter class  $P(C_h|H(i))$ , while the other model parameters were still estimated using Witten-Bell. We then extended the use of  $k$ -NN to include the parameter classes for generating modifying nonterminals. Collins treats the generation of nonterminals whose parent is a base noun phrase differently to all other modifying nonterminal generation. For the generation of modifying nonterminals there are therefore two different parameter classes, one for any constituent whose parent is an NPB,  $P(C_i, t_i, punc|C_p = NPB, H(i))$ , and another for all other constituents  $P(C_i, t_i, coord, punc|H(i))$ .<sup>3</sup> Finally, we further decomposed these two modifying non-terminal parameter classes.<sup>4</sup>

Tables 3.1, 3.2, and 3.3 outline the parameter classes estimated using  $k$ -NN in the final model settings and shows the feature sets used for each parameter class as well as the constraint feature settings. For the constraint feature columns, where a particular parameter class has more than one set of constraint features, the more restrictive set, that is the set containing more constraint features, will be used only when the training set contains a number of examples exceeding a certain threshold. Where we used more than one level of constraint features, we set the threshold over which the more restrictive constraint feature set was employed to 5000, and the minimum training set size threshold to 100. Table 3.1 gives the details of the parameter class for generating the head child node. Table 3.2 gives the details of most of the parameter classes used for the generation of modifier nodes. This table has a column *dir* which indicates that the parameter class is used for modifiers to the left of the head child (left) or to the right of the head child (right) or is used in both cases (left|right). The column  $C_p$  indicates which parameter classes are not used for nodes with NPB parents (!NPB). When there is no value for this column entry then the parameter class is used for all values of  $C_p$ . The final table, Table 3.3, gives the details of the remaining modifier parameter classes, used only for nodes whose parent is a base noun phrase.

We did not extend the use of  $k$ -NN estimation to include the other, minor, parameter classes in the complete model, which continue to be estimated using Witten-Bell.

---

<sup>3</sup>*coord* and *punc* denote the coordination and punctuation flags respectively.

<sup>4</sup>We found experimentally that decomposing the parameter classes seemed to work well for  $k$ -NN. We can hypothesise that  $k$ -NN for parameter estimation does better with class variables that have fewer possible values. This might also explain why  $k$ -NN alone did not work well for the parameter class  $P(w_i|H(i))$  which has a *word* class variable.



Head Nodes			
Parameter Class	Future	History	Constraint Features
$P(C_h H(i))$	$C_h$	$C_p, w_p, t_p, t_{gp}$	$\{C_p\}$

Table 3.1: The parameter class for generating  $C_h$ , the non-terminal label of the head child node.  $C_p$  is the parent non-terminal label,  $w_p$  and  $t_p$  its head word and part-of-speech respectively, and  $t_{gp}$  is the POS tag of the grandparent node.

Modifier Nodes					
dir	$C_p$	Parameter Class	Future	History	Constraint Features
left	!NPB	$P(t_i H(i))$	$t_i$	$dir, C_p, C_h, w_p, dist, t_{i-1}, t_{i-2}, C_{gp}$	$\{dir, C_p\}, \{dir, C_p, C_h\}$
		$P(w_i H(i))$	$w_i$	$t_i, C_i, C_p, C_h, w_p, dist, t_{i-1}, t_{i-2}, C_{gp}$	$\{t_i\}$
right	!NPB	$P(t_i H(i))$	$t_i$	$dir, C_p, C_h, w_p, t_p, dist, t_{i-1}, t_{i-2}, C_{gp}$	$\{dir, C_p\}, \{dir, C_p, C_h\}$
		$P(w_i H(i))$	$w_i$	$t_i, C_i, C_p, C_h, w_p, t_p, dist, t_{i-1}, t_{i-2}, C_{gp}$	$\{t_i\}$
left right	!NPB	$P(C_i H(i))$	$C_i$	$dir, t_i, C_p, C_h, w_p, t_p, dist, t_{i-1}, t_{i-2}, C_{gp}$	$\{dir, t_i\}, \{dir, t_i, C_p\}$
		$P(coord, punc H(i))$	$coord, punc$	$dir, C_i, t_i, C_p, C_h, w_p, t_p$	$\{dir, C_i, t_i\}$

Table 3.2: The parameter classes for the generation of modifier nodes. The notation is that used throughout the thesis.  $dir$  is a flag which indicates whether the modifier being generated is to the left or the right of the head child.  $dist$  is the distance metric used in the Collins parser.  $t_{i-1}$  and  $t_{i-2}$  are the POS tags for the previous two generated nodes.  $C_{gp}$  is the grandparent non-terminal label.

Modifier Nodes, $C_p = \text{NPB}$				
Parameter Class	Future	History		Constraint Features
$P(C_i, t_i H(i))$	$C_i, t_i$	$dir, C_p, C_h, w_p, C_{i-2}, w_{i-2}, C_{i-3}, w_{i-3}, C_{gp}, C_{ggp}, C_{gggp}$		$\{dir, C_p, C_h\}$
$P(punc H(i))$	$punc$	$dir, C_p, t_i, C_i, C_h, w_p, t_p, t_{i-1}, t_{i-2}$		$\{dir, C_p, t_i\}$

Table 3.3: The parameter classes used only when  $C_p = \text{NPB}$ . The notation is that used throughout the thesis. In addition,  $C_{ggp}$  and  $C_{gggp}$  are the great- and great-great-grandparent non-terminal labels respectively.  $C_{i-2}, w_{i-2}$  and  $C_{i-3}, w_{i-3}$  are the non-terminal labels and head words of the second and third previously generated nodes.

As noted earlier, we extend the original feature sets by increasing the order of both horizontal and vertical markovisation. From each constituent node in the vertical or horizontal history we chose features from among the constituent’s nonterminal label, its head word and the head word’s part-of-speech tag. As outlined in [Bikel, 2004a] for the modifying parameter classes where the parent node has an NPB label, the head child is taken to be the previously generated modifier, so  $C_h$  is always  $C_{i-1}$  for these parameter classes.

As well as parameter class decomposition and feature selection for each parameter class, fine-tuning the model involved choosing the best value for  $k$  for each of the pa-

$\leq 40$ Words(2245 Sentences), Section 23					
Model	LR	LP	CBs	0 CBs	2CBs
WB Baseline	88.2%	88.5%	0.93	65.66%	86.9%
CO99 M1	87.9%	88.2%	0.95	65.8%	86.3%
CO99 M2	88.5%	88.7%	0.92	66.7%	87.1%
Bikel 1-best	88.7%	88.7%	0.92	67.2%	86.95%
$k$ -NN	89.2%	89.6%	0.84	68.0%	88.2%

Table 3.4: Results for sentences of less than or equal to 40 words, from section 23 of the Penn treebank. LP/LR =Labelled Precision/Recall. CBs = the average number of Crossing Brackets per sentence. 0 CBs, 2 CBs are the percentage of sentences with 0 or  $\leq 2$  crossing brackets respectively. WB Baseline is our baseline emulation of Model 1 when tested on the output of the Bikel  $n$ -best parser. CO99 M1 and M2 are [Collins, 1999] Models 1 and 2 respectively. Bikel 1-best is [Bikel, 2004a].  $k$ -NN is our final  $k$ -NN model.

parameter classes estimated with  $k$ -NN and the best distance weighting function. Initially we experimented with different distant weighting functions. We tried both the inverse distance weighting function  $\frac{1}{(d+1)^m}$  and exponential decay  $e^{-\alpha d^{\beta}}$ , but found little difference between the two and settled on the inverse distance function, being marginally better. In initial experiments we also used a grid search: for different values of  $k$ , we kept  $k$  fixed and then altered the constant value in the inverse distance weighting function. We found for all parameter classes  $k = 10,000$  or  $k = 20,000$  worked best. The distance weighting function that worked best was the inverse distance weighting function, with settings of either  $(\frac{1}{(d+1)})^6$  or  $(\frac{1}{(d+1)})^7$ .

### 3.6 Results

The results are shown in Table 3.4. The higher scores we achieve with our emulation of Model 1 over Collins Model 1 are no doubt due to the fact that unlike Collins parser, which starts with (POS tagged) sentences, we take as input a set of  $n$ -best parses generated by a parser based on Model 2. Our Witten-Bell Model 1 scores are lower than Bikel’s 1-best score, presumably because the Bikel parser uses Model 2. With our  $k$ -NN model we achieve LR/LP of 89.2%/89.6% on sentences  $\leq 40$  words. These results show an 9% relative reduction in  $f$ -score error over our Model 1 baseline and

a 6% relative reduction in  $f$ -score error over the Bikel parser which, as the parser we used as our base parser, is the second baseline score against which we evaluate our results. See Table 2.1 for a summary of the performance of state-of-the-art parsers and discriminative rerankers on the same test set.

We compared the results of our  $k$ -NN model against the Bikel 1-best parser results using the paired T test where the data points being compared were the scores of each parse in the two different sets of parses. Following Collins [2000] the score of a parse,  $x$ , is calculated as follows:

$$Score(x) = \frac{fscore(x)}{100} * size(x)$$

where

$$fscore(x) = \frac{LR * LP}{.5(LR + LP)}$$

and  $size(x)$  is the number of constituents in the gold standard parse for this sentence. It is a good idea to take into account the size of the parse tree because it is relatively easy to get an  $f$ -score of, say, 100% on short sentences, and much harder on longer sentences, so accounting for the size when scoring a particular parse gives a more accurate picture of how well the model is doing. The mean scores for Bikel's 1-best and the  $k$ -NN model were 15.06 and 15.17 respectively. The mean difference between the scores was 0.12, with standard deviation 1.57. The 95% confidence interval for the mean difference between the scores of the paired sets of parses is [0.05, 0.18] with  $p < .0005$ .

Using  $k$ -NN for parameter estimation, enriching the feature sets used for prediction, in combination with a further decomposition of parameter classes produced significant improvements in parser accuracy over the original model with Witten-Bell estimation. These results show that using  $k$ -NN to estimate local probabilities in the generative parsing models presented in this chapter is highly effective for parameter estimation.  $k$ -NN allows for flexible feature selection and good smoothing of data and can achieve state-of-the-art results for accuracy.

### 3.7 Computational Costs

The  $k$ -NN algorithm for probability estimation delays generalising beyond the training data until it must assign a probability to a new query instance. While this lazy learning has advantages insofar as an estimate is custom-built for a new query instance, it carries with it the disadvantage that the computational cost of producing an estimate for a new query instance can be high. This is because nearly all computation takes place not when the training examples are first encountered but at run time. Though  $k$ -NN allows for greater flexibility in terms of conditioning features, the computational cost of estimating parameter classes with many features must be taken into consideration, with each extra feature contributing significantly to slowing down the speed of the parser.

All experiments in this thesis were carried out on a machine with a 1.7 GHz Intel Pentium processor and 1 GB of memory. Due to memory limitations, we carried out the experiments as follows: to rerank a set of  $n$ -best lists we calculated the estimates for each parameter class in separate runs. When all parameter class estimates needed to rerank the lists were calculated we ran the reranker with the pre-computed estimates. To rerank the 2245  $n$ -best lists for section 23, this process took in total 4 hours 10 minutes. All our experiments were focused on improving accuracy and we believe the program could be speeded up considerably with more attention paid to efficiency, as well as, obviously, a higher specification machine.

Given the computational cost of parameter estimation with  $k$ -NN, it would not be feasible to apply the memory-based model directly in a one-pass parser like that of Collins [1999]. However, applying the model as the second-pass of a two-pass parser, along the lines of Charniak [2000]'s parser, would be a worthwhile area of future research.

### 3.8 Relation to Previous Work

Our parsing model is related to previous work on history-based parsing, in particular the generative approaches of Collins [1999] and Charniak [2000]. Our baseline model is a direct replication of the Collins' parsing model, Model 1. In the development of the parsing model we depended on the details in Collins thesis as well as the previously



unpublished details outlined in Bikel [2004b]. Although our model is applied to the parse reranking task, it is more similar to the second-pass of a parser, such as the two-pass parser of Charniak [2000], than the discriminative rerankers of, for example, Collins [2000]. Our model differs from previous work in generative history-based parsing in the parameter estimation technique used (memory-based parameter estimation) and in the parameterisation of the parsing model.

Our memory-based parameter estimation technique is similar that used in [Toutanova et al., 2003]’s experiments with  $k$ -NN for parameter estimation in a HPSG parsing model. In their experiments they also train generative history-based models, but for HPSG derivation trees. The work presented in this chapter, however, differs from that of Toutanova et al. [2003], not only in our domain of application, but also in our use of constraint features, as well as our combination of  $k$ -NN estimates with Witten-Bell estimates as a way of smoothing lexical probabilities.

### 3.9 Conclusion

This chapter describes a generative parsing model which uses  $k$ -NN for local probability estimation. Taking the  $n$ -best output of a base parser and re-estimating the probability of each parse, it achieves an  $f$ -score of 89.4% which is a 6% relative error reduction over the 1-best output of the base parser. Although in our experiments we rerank the parses output from a base parser, our model differs from other rerankers in that it is generative and could conceivably be incorporated into a base parser as a second pass.

Discriminative rerankers have advantages over history-based approaches in that they are not restricted to choosing features from the parse derivation history but instead can use additional features which incorporate arbitrary aspects of the whole parse tree to improve the initial ranking of the base parser. In the discriminative rerankers of [Collins, 2000, Collins and Duffy, 2002, Shen et al., 2003], the log probability given by the base parser for each of the  $n$ -best parses of a sentence is used in the computation of the new score of a parse. Our generative model improves the ranking of an initial base parser by recalculating the log probability of each parse produced by the base parser and so produces a more accurate ranking of parses along with their log probabilities. It is possible that improving the log probability ranking of a base parser therefore could improve the scores of the discriminative reranker which uses these log probabilities in

its reranking algorithm. We discuss this further in Chapter 8 when we outline areas of future work.

# Chapter 4

## Conjoined Lexical Head Nouns

### 4.1 Introduction

In this chapter we begin by discussing how a measure that captures similarity of conjoined nouns might be useful for coordinate noun phrase disambiguation. This is our motivation for developing a measure of distributional similarity based on coordination patterns. We also look at other measures of word similarity based on WordNet and test all similarity measures on WSJ data to see if they can detect similarity in conjoined head nouns.

The second part of the chapter is devoted to modelling the likelihood of one noun being conjoined with another. We show how the head-head dependencies of coordinate noun phrases are not captured in the baseline model and develop a parameter class for the estimation of coordinate nouns in both noun phrases and base noun phrases. As this parameter class involves blexical statistics, data is extremely sparse and the remainder of the chapter is centred around improving the estimation of the parameter class by building a word graph from BNC and WSJ data and incorporating our word similarity measure into the parsing model.

### 4.2 Measures of Word Similarity

Some noun pairs are more likely to be conjoined than others. Take the trees in Figure 4.1 (the phrase is taken from the WSJ). The two head nouns coordinated in Tree 1

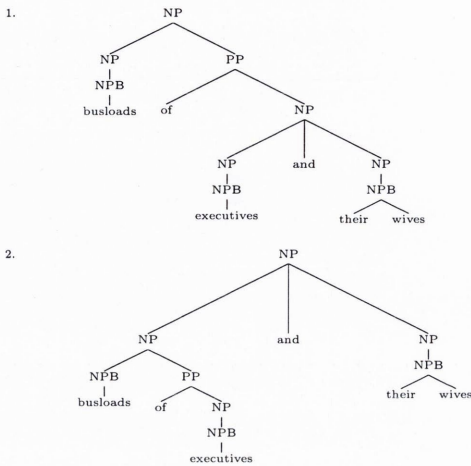


Figure 4.1: Tree 1. The correct noun phrase parse. Tree 2. The incorrect parse for the noun phrase.

are *executives* and *wives*, and in Tree 2: *busloads* and *wives*. Clearly, the former pair of head nouns is more likely and, for the purpose of discrimination, the model would benefit if it could learn that *executives and wives* is a more likely combination than *busloads and wives*. On inspection of noun phrase coordination in the data it seemed clear that nouns cooccurring in coordination patterns were often semantically similar, and therefore if a similarity measure could be defined so that, for example:

$$\text{sim}(\textit{executives}, \textit{wives}) > \text{sim}(\textit{busloads}, \textit{wives})$$

then it could be useful for coordination disambiguation.

The idea that nouns cooccurring in conjunctions tend to be semantically related has been noted in [Riloff and Shepherd, 1997] and used effectively to automatically cluster semantically similar words [Roark and Charniak, 1998, Caraballo, 1999, Widows and Dorow, 2002]. The tendency for conjoined nouns to be semantically similar has also been exploited for coordinate noun phrase disambiguation by Resnik [1999] who employed a measure of similarity based on WordNet to measure which were the head nouns being conjoined in certain types of coordinate noun phrase.



Semantic similarity can be defined in different ways, from a very narrow definition of word similarity that would define words as similar only if they are synonyms, to definitions of similarity based on membership of the same semantic category (e.g. *apples*, *pears*, and *bananas* are all members of the category *fruit*) or, even more generally, similarity based on a more general notion of semantic relatedness, where words are similar if they are related in meaning somehow (such as *car* and *wheel*).

A number of measures of semantic similarity have been developed which are based on similarity in lexical taxonomies such as WordNet [Fellbaum, 1998]. Other measures of word similarity are based on distributional similarity: the idea that words which are semantically similar occur in similar contexts and with similar distributions. Context can be defined in terms of grammatical dependency relations (such as conjunctions or modifier-head dependencies), documents, or a window of context words around the word under consideration.

We now look at different measures of word similarity in order to discover whether they can detect empirically a tendency for conjoined nouns to be more similar than nouns which co-occur but are not conjoined. In 4.2.1 we introduce our measure of word similarity based on word vectors and in 4.2.2 we briefly describe some WordNet similarity measures which, in addition to our word vector measure, will be tested in our experiments in 4.2.3.

#### 4.2.1 Similarity based on Coordination Cooccurrences

The potential usefulness of a similarity measure depends on the particular application. An obvious place to start, when looking at similarity functions for measuring the type of semantic similarity common for coordinate nouns, would be a similarity function based on distributional similarity with context defined in terms of coordination patterns. Our measure of similarity is based on noun co-occurrence information, extracted from conjunctions and lists. We collected co-occurrence data on 82,579 distinct word types from the BNC and the hand-annotated WSJ. The exact details of how the co-occurrence data is extracted from the BNC and the WSJ and stored in a word graph is described later in Section 4.3.2. From the co-occurrence data we construct word vectors. Every dimension of a word vector represents another word type and the values of the components of the vector, the term weights, are derived from the coordinate

word co-occurrence counts. We used dampened co-occurrence counts, of the form:  $1 + \log(\text{count})$ , as the term weights for the word vectors. We used no pruning of word types, thus vectors had 82,579 dimensions. To measure the similarity of two words,  $w_1$  and  $w_2$ , we calculate the cosine of the angle between the two word vectors,  $\vec{w}_1$  and  $\vec{w}_2$ , as follows:

$$\cos(\vec{w}_1, \vec{w}_2) = \frac{\vec{w}_1 \cdot \vec{w}_2}{\|\vec{w}_1\| \|\vec{w}_2\|} \quad (4.1)$$

## 4.2.2 WordNet-Based Similarity Measures

We also examine the following measures of semantic similarity which are WordNet-based.<sup>1</sup> Wu and Palmer [1994] propose a measure of similarity of two concepts  $c_1$  and  $c_2$  based on the depth of concepts in the WordNet hierarchy. Similarity is measured from the depth of the most specific node dominating both  $c_1$  and  $c_2$ , (their lowest common subsumer), and normalised by the depths of  $c_1$  and  $c_2$ . In Resnik [1995] concepts in WordNet are augmented by corpus statistics and an information-theoretic measure of semantic similarity is calculated. Similarity of two concepts is measured by the information content of their lowest common subsumer in the *is-a* hierarchy of WordNet. Both Jiang and Conrath [1997] and Lin [1998] propose extensions of Resnik's measure. Leacock and Chodorow [1998]'s measure takes into account the path length between two concepts, which is scaled by the depth of the hierarchy in which they reside. In [Hirst and St-Onge, 1998] similarity is based on path length as well as the number of changes in the direction in the path. In [Banerjee and Pedersen, 2003] semantic relatedness between two concepts is based on the number of shared words in their WordNet definitions (glosses). The gloss of a particular concept is extended to include the glosses of other concepts to which it is related in the WordNet hierarchy. Finally, Patwardhan and Pedersen [2006] build on previous work on second-order cooccurrence vectors [Schütze, 1998] by constructing second-order co-occurrence

---

<sup>1</sup>All of the WordNet-based similarity measure experiments, as well as a random similarity measure, were carried out with the WordNet::Similarity package, which is freely available for download from <http://search.cpan.org/dist/WordNet-Similarity>.

SimTest	Authors	Description
coordGraph	this work	cosine similarity derived from coordination co-occurrences
res	[Resnik, 1995]	WordNet-based, information theoretic
lin	[Lin, 1998]	WordNet-based, information theoretic
jcn	[Jiang and Conrath, 1997]	WordNet-based, information theoretic
wup	[Wu and Palmer, 1994]	WordNet-based, depth in WordNet hierarchy
lch	[Leacock and Chodorow, 1998]	WordNet-based, path length and depth in hierarchy
hso	[Hirst and St-Onge, 1998]	WordNet-based, path length and number of changes in direction
lesk	[Banerjee and Pedersen, 2003]	number of shared words in WordNet glosses
vectorGloss	[Patwardhan and Pedersen, 2006]	similarity of 2nd order cooccurrence vectors derived from WordNet glosses

Table 4.1: Summary of the 9 different word similarity measures to be evaluated empirically on WSJ cooccurrence data.

vectors from WordNet glosses, where, as in [Banerjee and Pedersen, 2003], the gloss of a concept is extended so that it includes the gloss of concepts to which it is directly related in WordNet.

### 4.2.3 Empirical Evaluation of Similarity Measures

We selected two sets of data from sections 00, 01, 22 and 24 of the WSJ treebank. (The WSJ data used to train our vector similarity function is from sections 02 to 21.) The first consists of all nouns pairs which make up the head words of two conjuncts in coordinate noun phrases (detected when the coordination flag is set to true, and therefore not including coordinate NPBs). We found 601 such coordinate noun pairs. The second data set consists of 601 word pairs which were selected at random from all head-modifier pairs in the same sections of the WSJ where both head and modifier words are nouns and are *not* coordinated. We tested the 9 different measures of word similarity just described and summarised in Table 4.1 on each data set in order to see if, through using the measures, a significant difference could be detected between the similarity scores for the coordinate words sample and non-coordinate words sample.

Initially both the coordinate and non-coordinate pair samples each contained 601 word pairs. However, before running the experiments we removed from the sets all pairs where the words in the pair were identical. This is because identical words occur more often in coordinate head words than in other lexical dependencies (there were 43 pairs where the two words in the pair were identical in the coordination set, compared



SimTest	$n_{coord}$	$\bar{x}_{coord}$	$SD_{coord}$	$n_{nonCoord}$	$\bar{x}_{nonCoord}$	$SD_{nonCoord}$	95% CI	p-value
coordGraph	503	0.11	0.13	485	0.06	0.09	[0.04 0.07]	0.000
res	444	3.19	2.33	396	2.43	2.10	[0.46 1.06]	0.000
lin	444	0.27	0.26	396	0.19	0.22	[0.04 0.11]	0.000
jcn	444	0.13	0.65	395	0.07	0.08	[-0.01 0.11]	0.083
wup	444	0.63	0.19	396	0.55	0.19	[0.06 0.11]	0.000
lch	444	1.72	0.51	396	1.52	0.47	[0.13 0.27]	0.000
hso	459	1.599	2.03	447	1.09	1.87	[0.25 0.76]	0.000
lesk	451	114.12	317.18	436	82.20	168.21	[-1.08 64.92]	0.058
vectorGloss	459	0.67	0.18	447	0.66	0.2	[-0.02 0.03]	0.545
random	483	0.89	0.17	447	0.88	0.18	[-0.02 0.02]	0.859

Table 4.2: Summary statistics for 9 different word similarity measures (plus one random measure):  $n_{coord}$  and  $n_{nonCoord}$  are the sample sizes for the coordinate and non-coordinate noun pairs samples, respectively;  $\bar{x}_{coord}$ ,  $SD_{coord}$  and  $\bar{x}_{nonCoord}$ ,  $SD_{nonCoord}$  are the sample means and standard deviations for the two sets. The 95% CI column shows the 95% confidence interval for the difference between the two sample means. The p-value is for a Welch two sample two-sided t-test.

to 3 such pairs in the non-coordination set). If we had not removed them, a statistically significant difference between the similarity scores of the pairs in the two sets could be found simply by using a measure which, say, gave one score for identical words and another (lower) score for all non-identical word pairs.

Results for all tests on the data sets described above are displayed in Table 4.2. The similarity measures displayed are: (coordGraph) our vector similarity described above in 4.2.1, and (res) [Resnik, 1995], (lin) [Lin, 1998], (jcn) [Jiang and Conrath, 1997], (wup) [Wu and Palmer, 1994], (lch) [Leacock and Chodorow, 1998], (hso) [Hirst and St-Onge, 1998], (lesk) [Banerjee and Pedersen, 2003] and (vectorGloss) Patwardhan and Pedersen [2006]. In one final experiment we used a random measure of similarity. For each experiment we produced two samples, one consisting of the similarity scores given by the similarity measure for the coordinate noun pairs, and another set of similarity scores generated for the non-coordinate pairs. The sample sizes, means, and standard deviations for each experiment are shown in the table. Note that the variation in the sample size is due to coverage: the different measures did not produce a score for all word pairs. Also displayed in Table 4.2 are the results of statistical significance tests based on the Welch two sample t-test. A 95% confidence interval for the difference of the sample means is shown along with the p-value.



#### 4.2.4 Discussion

For all but three of the experiments (not including the random measure), the difference between the mean similarity measures is statistically significant. Interestingly, the three tests where no significant difference was measured between the scores on the coordination set and the non-coordination set [Jiang and Conrath, 1997, Banerjee and Pedersen, 2003, Patwardhan and Pedersen, 2006] were the three top scoring measures in [Patwardhan and Pedersen, 2006], where a subset of 6 of the above WordNet-based experiments were compared and the measures evaluated against human relatedness judgements and in a word sense disambiguation task. In another comparative study [Budanitsky and Hirst, 2002] of five of the above WordNet-based measures, evaluated as part of a real-word spelling correction system, Jiang and Conrath [1997]’s similarity score performed best. Although performing relatively well under other evaluation criteria, these three measures seem less suited to measuring the kind of similarity occurring in coordinate noun pairs. One possible explanation for the unsuitability of the measures of [Patwardhan and Pedersen, 2006] for the coordinate similarity task could be based on how context is defined for the building of their context vectors. Context for an instance of the the word  $w$  is taken to be the words (minus low frequency and stop words) that surround  $w$  in the corpus within a given number of positions, where the corpus is taken as all the glosses in WordNet. Words that form part of collocations such as *disk drives* or *task force* would then tend to have very similar contexts, and thus such word pairs, from non-coordinate modifier-head relations, could be given too high a similarity score.

Although the difference between the mean similarity scores seems rather slight in all experiments, it is worth noting that not all coordinate head words *are* semantically related. To take a couple of examples from the coordinate word pair set: *work/harmony* extracted from *hard work and harmony*, and *power/clause* extracted from *executive power and the appropriations clause*. We would not expect these word pairs to get a high similarity score. On the other hand, it is also possible that some of the examples of non-coordinate dependencies involve semantically similar words. For example, nouns in lists are often semantically similar, and we did not exclude nouns extracted from lists from the non-coordinate test set.

Although not all coordinate noun pairs are semantically similar, it seems clear, on inspection of the two sets of data, that they are more likely to be semantically similar

than modifier-head word pairs, and the tests carried out for most of the measures of semantic similarity detect a significant difference between the similarity scores assigned to coordinate pairs and those assigned to non-coordinate pairs. The measure of distributional similarity introduced in Section 4.2.1 also measured significant differences between the two data sets.

It is not possible to judge, based on the significance tests alone, which might be the most useful measure for the purpose of disambiguation. However, in terms of coverage, the coordinate word graph measure clearly performs best (somewhat unsurprisingly given it is part trained on data from the same domain). This measure of distributional similarity is perhaps more suited to the task of coordination disambiguation because it directly measures the type of similarity that occurs between coordinate nouns. That is, the distributional similarity measure presented in Section 4.2.1 defines two words as similar if they occur in coordination patterns with a similar set of words and with similar distributions. Whether, or to what degree, the words are *semantically* similar becomes irrelevant. A measure of semantic similarity, on the other hand, might find words similar which are quite unlikely to appear in coordination patterns. For example, Cederberg and Widdows [2003] note that words appearing in coordination patterns tend to be on the same ontological level: ‘fruit and vegetables’ is quite likely to occur, whereas ‘fruit and apples’ is an unlikely cooccurrence. A WordNet-based measure of semantic similarity, however, might give a high score to both of the noun pairs.

In the next section, we look at how best to model the dependencies between coordinate head words in the parsing model, and show how the coordinate word graph similarity measure might be incorporated into the parsing model to aid noun phrase coordination disambiguation.

### 4.3 Modelling Coordinate Head Words

Bilexical head-head dependencies of the type found in coordinate structures are a somewhat different class of dependency to modifier-head dependencies. In *the fat cat*, for example, there is clearly one head to the noun phrase: *cat*. In *cats and dogs* however there are two heads, though in the parsing model just one is chosen, somewhat arbitrarily, to head the entire noun phrase.

In the baseline model there is essentially one parameter class for the estimation of

word probabilities:

$$P_{word}(w_i|H(i)) \quad (4.2)$$

where  $w_i$  is the lexical head of constituent  $i$  and  $H(i)$  is the *history* of the constituent. The *history* is made up of conditioning features chosen from structure that has already been determined in the top-down derivation of the tree.

For certain types of coordinate NP, such as the coordinate noun phrases of Figure 4.1, the head-head dependency is captured in the model when one feature of the history, the coordination flag, is set to true. Parses are generated top-down, head-first, left-to-right. For the trees in Figure 4.1, discarding for simplicity the other features in the history, the probability of the coordinate head *wives*, is estimated in Tree 1 as:

$$P_{word}(w_i = \textit{wives} | coord = true, w_p = \textit{executives}, \dots) \quad (4.3)$$

and in Tree 2:

$$P_{word}(w_i = \textit{wives} | coord = true, w_p = \textit{busloads}, \dots) \quad (4.4)$$

where  $w_p$  is the head word of the node to which the node headed by  $w_i$  is attaching and *coord* is the coordination flag. However, as we discuss further in the next section, for NPBs this coordinate head-head dependency is not captured in the probability model.

In Section 4.3.1 we look at how we might improve the model's handling of coordinate head-head dependencies by altering the model so that the common parameter class in (4.5) is used for coordinate word probability estimation in both noun phrases and base noun phrases.

$$P_{coordWord}(w_i|w_p, H(i)) \quad (4.5)$$

In Section 4.3.2 we focus on improving the estimation of this parameter class by including BNC data to reduce data sparseness.

### 4.3.1 Extending $P_{coordWord}$ to Coordinate NPBs

As described in §1.3, coordination in base NPs is handled differently to coordination in NPs in the Collins model. Unlike NPs, in NPBs (i.e. flat, non-recursive NPs) the coordination flag is not used to mark whether a node is a coordinated head or not. This flag is always set to false for NPBs. In addition, unlike other NPs, modifiers within NPBs are conditioned on the previously generated modifier rather than the head of the phrase.<sup>2</sup> This means that, in the baseline model, in an NPB such as (*cats and dogs*), the estimate for the word *cats* will look like:

$$P_{word}(w_i = \textit{cats} | coord = \textit{false}, w_p = \textit{and}, \dots) \quad (4.6)$$

We alter the baseline model so that, for NPs, when the coordination flag is set to true, we use the parameter class in (4.5) to estimate the probability of one lexical head noun, given another. In order to capture head-head dependencies in coordinate NPBs, if a noun is generated directly after a coordinating conjunction in an NPB then it is taken to be a coordinate head,  $w_i$ , and conditioned on the noun generated before the coordinating conjunction (which is taken to be  $w_p$ ) and also estimated using (4.5).

### 4.3.2 Estimating the $P_{coordWord}$ Parameter Class from a Coordination Word Graph

#### Building the Word Graph

Data for bilexical statistics are particularly sparse. In order to decrease the sparseness of the coordinate head noun data, we extracted from the BNC examples of coordinate head noun pairs. We extracted all noun pairs occurring in a pattern of the form: *noun cc noun*, as well as lists of any number of nouns separated by commas and ending in *cc noun* (note these are not the actual BNC tags). To the BNC data we added all head noun pairs from the WSJ (sections 02 to 21) that occurred together in a coordinate noun phrase, identified when the coordination flag was set to true. We did not include coordinate head nouns from NPBs because the underspecified annotation of NPBs in the WSJ means that the conjoined head nouns can not always be automatically identified. We stored these coordinate head noun samples in a graph, where each vertex

---

<sup>2</sup>A full explanation of the handling of coordination in the model is given in [Bikel, 2004a].



in the graph represents a word and the edges between vertices indicate the number of times two words have occurred together in a coordination pattern in the training set. The graph is undirected; thus an occurrence, say, of ‘apples and bananas’ also counts as an occurrence of ‘bananas and apples’. This further helps reduce sparseness. For lists of nouns, each noun in the list is linked with every other noun in the list. Thus for a list:  $n_1$ ,  $n_2$ , and  $n_3$ , there will be links between nodes  $n_1$  and  $n_2$ , between  $n_1$  and  $n_3$  and between  $n_2$  and  $n_3$ . Therefore, for each list or coordinate pair extracted from the corpora, containing  $m$  nouns, there are  $m^2 - m$  coordination events.

To illustrate, take the following examples extracted from the BNC:

*teachers, nurses, engineers and mariners*  
*doctors, nurses and teachers*  
*doctors and nurses*  
*sailors and mariners*

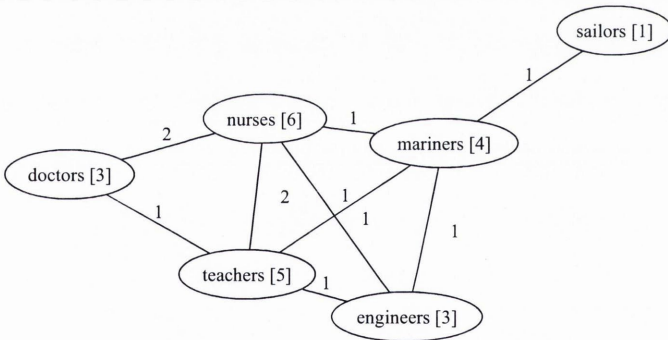


Figure 4.2: Graph of coordinations extracted from the BNC.

These examples are used to construct the graph in Figure 4.2. In total the graph stores the equivalent of 22 coordination events or training samples where each sample consists of two words, one the class value and the other the sole feature value (the history).

## Estimation from the Word Graph

The probability of one noun  $n_i$  being coordinated with another  $n_j$  can be calculated simply from the graph as:

$$P_{WG}(n_i|n_j) = \frac{\text{count}(n_i n_j)}{\text{count}(n_j)} \quad (4.7)$$

where  $\text{count}(n_i n_j)$  is the number of links between node  $n_i$  and  $n_j$  in the graph, and  $\text{count}(n_j)$  is the count of word token  $n_j$  in the graph. This is just the unsmoothed maximum likelihood estimate. Again to reduce data sparseness, we introduce a measure of word similarity so that if we have examples of *bananas* and *apples* in the graph but no examples of *oranges* and *apples*, for which we need an estimate, it would be useful to know that *bananas* are similar to *oranges*, and as we've seen examples of *bananas* and *apples*, then an occurrence of *oranges* and *apples* should be reasonably likely.

We use the measure of word similarity described in §4.2, which is based on cosine similarity of word vectors derived from the graph of coordinate words (equation (4.1)). The graph representation lends itself easily to measures of word similarity based on vectors. A word is represented as a vector where every word in the graph is a dimension of the vector. In this case the values of the vector components are derived from the number of links between the two word nodes in the graph (i.e. the number of times the words occurred together in a coordination pattern in the training set). Note that in Section 4.2 we measured similarity between two, possibly coordinate, head words,  $noun_i$  and  $noun_j$ . Here, however, we do not directly measure the similarity of the two head words but rather use the similarity function both to find words similar to  $noun_j$  to include in the sample when estimating the probability of  $noun_i$  being conjoined with  $noun_j$ , and as a weight to determine the contribution of these similar words during estimation.

We alter the probability estimate of (4.7) to incorporate the similarity measure in the following manner:

$$P_{sim}(n_i|n_j) = \frac{\sum_{n_x \in N(n_j)} \text{sim}(n_j, n_x) \text{count}(n_i n_x)}{\sum_{n_x \in N(n_j)} \text{sim}(n_j, n_x) \text{count}(n_x)} \quad (4.8)$$

where  $\text{sim}(n_j, n_x)$  is a similarity score between words  $n_j$  and  $n_x$  and  $N(n_j)$  is the set of words in the neighbourhood of  $n_j$ . This neighbourhood can be based on the

$k$ -nearest neighbours of  $n_j$ , where nearness is measured with the similarity function.

The estimate in (4.8) can be viewed as the estimate with the more general history context because the context will include not only  $n_j$  but also words similar to  $n_j$ . We combine the estimate in (4.7) with the more general estimate in (4.8) by way of linear interpolation so that the final estimate is calculated as follows:

$$P_{coordWord}(n_i|n_j) = \lambda_{n_j} P_{WG}(n_i|n_j) + (1 - \lambda_{n_j}) P_{simInterp}(n_i|n_j) \quad (4.9)$$

where

$$P_{simInterp}(n_i|n_j) = \lambda'_{n_j} P_{sim}(n_i|n_j) + (1 - \lambda'_{n_j}) P_{kNN}(n_i|H(i)) \quad (4.10)$$

$P_{kNN}(n_i|H(i))$  is a  $k$ -NN estimate calculated in the same fashion as the  $k$ -NN estimates described in chapter 3. We include this final layer of backoff in order to smooth the bilexical estimate further.  $\lambda_{n_j}$  is calculated using Witten-Bell interpolation and so the linear combination of estimates is similar to how the maximum-likelihood estimates were combined for smoothing in the Collins parser. However for the calculation of the weight  $\lambda'_{n_j}$  in (4.10) we adapt the Witten-Bell method so that it incorporates the similarity measure for all words in the neighbourhood of  $n_j$ , as follows:

$$\lambda_{n_j} = \frac{\sum_{n_x \in N(n_j)} sim(n_j, n_x) count(n_x)}{\sum_{n_x \in N(n_j)} sim(n_j, n_x) (count(n_x) + CD(n_x))} \quad (4.11)$$

where  $C$  is a constant that can be optimised using held-out data and  $D(n_j)$  is the diversity of a word  $n_j$ . The diversity of a word is the number of distinct words (i.e. word *types*) with which  $n_j$  has been coordinated in the training set.

## 4.4 Relation to Previous Work

In Section 4.3.2 we described how a word graph was built from noun coordination and list cooccurrence statistics. Word cooccurrence statistics, based on coordination and list contexts, have been used successfully in the past to extract semantic information for building semantic lexicons [Roark and Charniak, 1998, Widdows and Dorow, 2002] and to improve recall in automatic hyponymy extraction [Cederberg and Widdows,

2003]. In [Roark and Charniak, 1998] the method for extracting co-occurring head nouns differs from ours in that the corpus (they use the MUC-4 and WSJ corpora) was first automatically parsed and the heuristics for extracting the head nouns were slightly different, including not just lists and conjunctions but also appositives (such as: the *plane*, a twin-engined *Cessna*)<sup>3</sup>. Building a graph of coordinate words, extracted from the BNC in the fashion described in §4.3.2, follows the work on graph-based models described in [Widdows and Dorow, 2002, Widdows, 2004]. Our graph additionally includes coordinate head nouns from the manually-annotated WSJ.

The graph we construct, partially from unparsed BNC data, is used to store events for the estimation of coordinate word probabilities. In the probabilistic approach to noun phrase coordination disambiguation presented in [Goldberg, 1999], unannotated data is also used. The model is trained from examples of unambiguous coordination noun phrases, of the form *n1 cc n2*, extracted from the unannotated WSJ. Their method of data collection differs from ours in several ways. For example, before searching for coordination patterns their unannotated data is chunked automatically using a simple chunker which replaces noun and quantifier phrases with their head words, whereas we collect data directly from the POS tagged BNC. Also the heuristics for collecting the *n1 cc n2* in Goldberg [1999] are more complicated than in our collection of data in that Goldberg puts more restrictions on the events collected, with the aim of only collecting unambiguous samples.

In §4.2.1 we introduce a measure of distributional similarity based on coordination patterns in the BNC and WSJ. In Nakov and Hearst [2005]’s approach to noun phrase coordination, the Web is used to get statistics on coordinate head word cooccurrences, though a similarity measure is not developed. Resnik [1999] also uses a measure of similarity to aid in noun phrase disambiguation but it is WordNet-based rather than based on coordination cooccurrences.

Measures of similarity between words based on similarity of cooccurrence vectors have been used for word sense disambiguation [Schütze, 1998, Patwardhan and Pedersen, 2006], for PP-attachment disambiguation [Zhao and Lin, 2004] and for the automatic construction of noun hierarchies [Caraballo, 1999]. Our approach resembles that of [Caraballo, 1999] where cooccurrence is also defined with respect to coordination patterns, although the experimental details in terms of data collection and vector term

---

<sup>3</sup>Example taken from [Roark and Charniak, 1998]



weights differ.

Our incorporation of a similarity measure into a probability estimate in (4.8) comes from  $k$ -NN estimation but bears some resemblance to the cooccurrence smoothing reviewed in §2.5.2. For the sake of comparison, we write the estimate in (4.8) as:

$$P_{sim.kNN}(w_2|w_1) = \frac{\sum_{w'_1 \in S(w_1)} sim(w_1, w'_1) |w_2 w'_1|}{\sum_{w'_1 \in S(w_1)} sim(w_1, w'_1) |w'_1|} \quad (4.12)$$

where  $sim(w_1, w'_1)$  is a similarity score between words  $w_1$  and  $w'_1$  and  $S(w_1)$  is the set of words in the neighbourhood of  $w_1$ .

In cooccurrence smoothing the form of (2.7) in §2.5.2 can be written as:

$$P_{sim.cooccur}(w_2|w_1) = \frac{\sum_{w'_1 \in S(w_1)} sim(w_1, w'_1) P(w_2|w'_1)}{\sum_{w'_1 \in S(w_1)} sim(w_1, w'_1)} \quad (4.13)$$

In cooccurrence smoothing the smoothed estimate is based on similarity-weighted *probability estimates*; our estimate, derived from  $k$ -NN, is based on similarity-weighted *events*.

## 4.5 Summary

Nouns that occur together in a coordination pattern are often semantically similar. We show that this can be detected by various different measures of semantic similarity. We also show how a measure of distributional similarity based on coordination patterns can also detect significant differences between the similarity of conjoined nouns and nouns that cooccur but are not conjoined. We argue that this latter measure is more suited to coordinate noun phrase disambiguation than WordNet-based measures of semantic relatedness.

We also show how the dependencies between conjoined head nouns are not adequately modelled in the baseline model and suggest an alternative that attempts to capture head-head dependencies in both NPs and base NPs. In order to improve the parameter estimation involving conjoined head nouns we build a word graph from both BNC and WSJ data and use the data stored therein for estimation. Finally, we show how a word similarity measure derived from the word graph data can be incorporated into the estimation of the head-head parameter class. In chapter 7 we show the effect

of changes suggested in this chapter on the baseline model.

# Chapter 5

## Parallelism Across Conjuncts

### 5.1 Introduction

In this chapter we carry out empirical measurements on coordination data from the WSJ in order to gauge the extent to which parallelism exists in the syntactic structure of two conjuncts. We then suggest an approach for altering the base parsing model so that it can capture a bias toward symmetry in conjunct structure, with the aim of improving coordination disambiguation accuracy.

### 5.2 Empirical Measurements of Parallelism

There is often a considerable amount of symmetry or parallelism in the syntactic structure of two conjuncts. Take Figure 5.1: If we take as level 0 the level in the coordinate sub-tree where the coordinating conjunction *CC* occurs, then there is exact symmetry in the two conjuncts in terms of non-terminal labels and headword part-of-speech tags for levels 0, 1 and 2.

In order to measure empirically the extent of parallelism across conjuncts we follow the work of Church [2000] on lexical priming and Dubey et al. [2005] on syntactic priming and parallelism in coordination, which we discuss in more detail in §5.4.

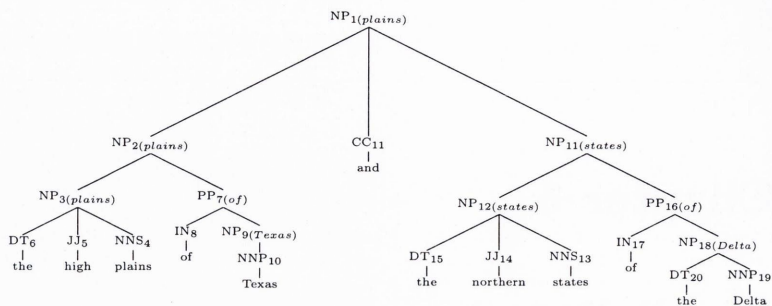


Figure 5.1: Example of symmetry in conjunct structure in a lexicalised subtree.

## 5.2.1 Methodology

We measure symmetry of conjunct structure in the training data based on counts of how often the syntactic labels in a post-CC conjunct of a coordinate phrase match the corresponding labels in the pre-CC conjunct. We compare the prior probability of a particular label occurring in the post-CC conjunct, with the probability of the label occurring in post-CC conjunct given it has occurred in the pre-CC conjunct. These correspond to the prior probabilities and *positive adaption* probabilities described in Church [2000]. We examine symmetry in conjunct structure across all conjunct types, with the exception of flat NPB constructions.

We first align each node,  $N_i$ , in the second conjunct with its corresponding pre-CC conjunct node,  $N_{i_{preCC}}$ . Note that when the structure of two conjuncts is different not all nodes in the post-CC conjunct will have a corresponding pre-CC conjunct node. For each node,  $N_i$ , in the post-CC conjunct we either align it with  $N_{i_{preCC}}$  or record that there is no corresponding node for  $N_i$  in the pre-CC conjunct.<sup>1</sup> When retrieving  $N_{i_{preCC}}$  for a post-CC conjunct node we tried both a left-to-right and a head-first traversal of the head conjunct.<sup>2,3</sup> The traversal of the pre-CC conjunct is guided by the position of  $N_i$  in the post-CC conjunct. For example, in a head-first traversal we

<sup>1</sup>We do not collect data on nodes in the pre-CC conjunct that have no corresponding node in the post-CC conjunct.

<sup>2</sup>Note that the head conjunct is always the pre-CC conjunct in the Collins model (with the exception of coordinate NPBs).

<sup>3</sup>See Zhang and Shasha [1989], for example, for other approaches to tree alignment, such as the tree-distance approach.



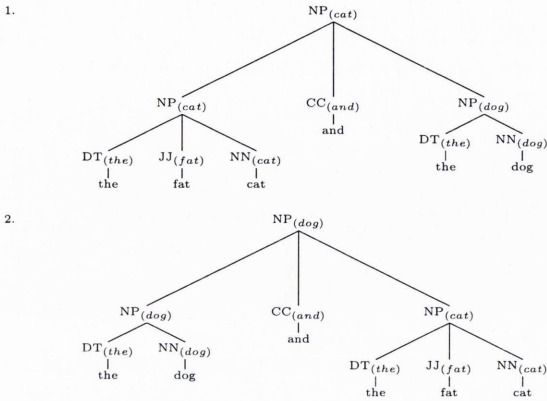


Figure 5.2: Trees that contain conjuncts with non-isomorphic structure.

align the head node at level 1 of the post-CC conjunct, with the head node of level 1 in the pre-CC conjunct; the first modifier node to the left of the head in the post-CC conjunct is aligned with the corresponding node in the pre-CC conjunct and so on. In a left-to-right alignment the left-most node at a particular level of the post-CC conjunct is aligned with the left-most node at the same level in the head conjunct; the second node from the left of post-CC conjunct is aligned with the second from the left in the head conjunct and so on.

While the head-first and left-to-right methods of aligning nodes are simple and reasonably effective, they will not always capture the full extent of symmetry in some non-isomorphic structures. For example, for the trees in Figure 5.2 the pairs of aligned nodes at level 1 are displayed in Table 5.1.

		pre-CC	post-CC
Tree 1	Head-First	NN(cat) JJ(fat)	NN(dog) DT(the)
	Left-to-Right	DT(the) JJ(fat)	DT(the) NN(dog)
Tree 2	Head-First	NN(dog) DT(the)	NN(cat) JJ(fat) DT(the)
	Left-to-Right	DT(the) NN(dog)	DT(the) JJ(fat) NN(cat)

Table 5.1: Nodes aligned at level 1 for the trees in Figure 5.2

	<i>test</i>	$\overline{test}$
<i>history</i>	$a = 136$	$b = 8$
$\overline{history}$	$c = 5$	$d = 13939$

Table 5.2: Contingency table for the head child non-terminal label  $TO$  at conjunct depth 1.

For our first set of experiments we use the set of aligned nodes to create lists of history-test pairs, to use the terminology of Church [2000]; the history samples coming from pre-CC conjuncts and test samples from post-CC conjuncts.

The first node to be generated in the expansion of a non-terminal is the head child node of the non-terminal, with label  $C_h$ . We first collected, via a head-first traversal, a set of history-test pairs of head child nodes in conjuncts at depth 1. For each distinct non-terminal label we estimated the prior probabilities and positive adaption probabilities. Following Church [2000], prior and positive adaption probabilities are calculated in the following manner. Take Table 5.2 which displays a contingency table for the non-terminal label  $TO$ , with counts collected from the history-test pairs for depth 1 head conjunct nodes.

The table shows that there are (a) 136 examples where  $TO$  is the head conjunct label in both test and aligned history node, there are (b) 8 examples where  $TO$  is the head conjunct label in the history node but not the test node, (c) 5 cases where  $TO$  occurs in the test but not the history node and finally (d) 13939 cases where  $TO$  is the head conjunct label in neither history nor test nodes. Positive adaption,  $P_{+adapt}$ , and prior,  $P_{prior}$ , probabilities are calculated as:

$$P_{+adapt} = P(C_{h_{test}} = TO | C_{h_{history}} = TO) = \frac{a}{a+b} \approx 0.94$$

$$P_{prior} = P(C_{h_{test}} = TO) = \frac{a+c}{a+b+c+d} \approx 0.01$$

As in [Dubey et al., 2005] whether  $TO$  occurring in the test set is independent of  $TO$  occurring in the history set can be tested using the  $\chi^2$  test for significance on the contingency table.

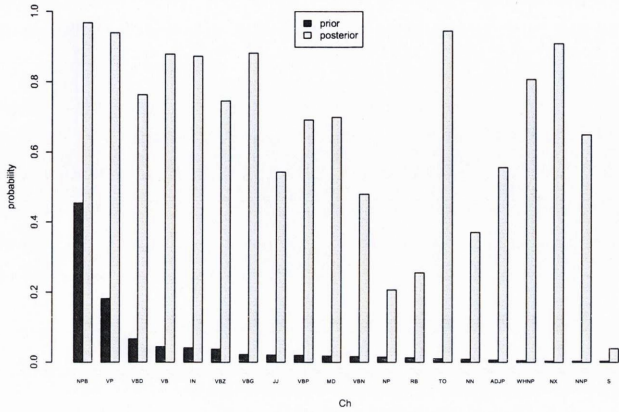


Figure 5.3: Prior and posterior (positive adaption) probabilities for head child non-terminal labels at conjunct depth 1

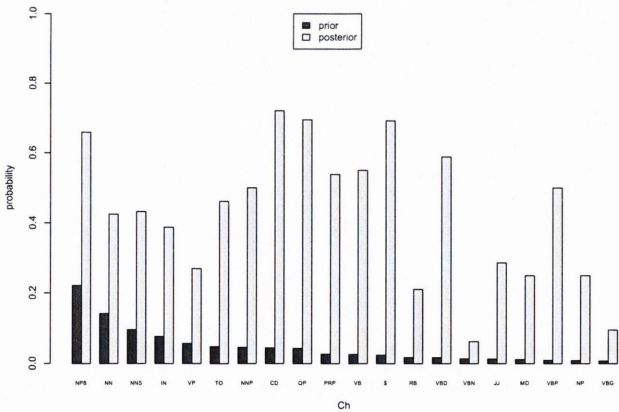


Figure 5.4: Prior and posterior (positive adaption) probabilities for head child non-terminal labels at conjunct depth 5

## 5.2.2 Results

The positive adaption and prior probabilities for the twenty most frequent  $C_h$  labels at this depth are displayed in Figure 5.3. Out of a total of 40 non-terminal label types for which we gathered statistics, in all cases the prior were less than the positive adaption probabilities. The difference in probabilities was statistically significant ( $p < 0.0001$ ) for 33 non-terminal types. We found that for most cases  $P_{+adapt} \gg P_{prior}$  was also true for depths greater than one, though the difference in prior and posterior probabilities reduced the greater the depth. Figure 5.4 displays the twenty most frequent  $C_h$  labels at conjunct depth 5. For a history-test pair set collected via a left-to-right traversal of the first conjunct we found similar results.

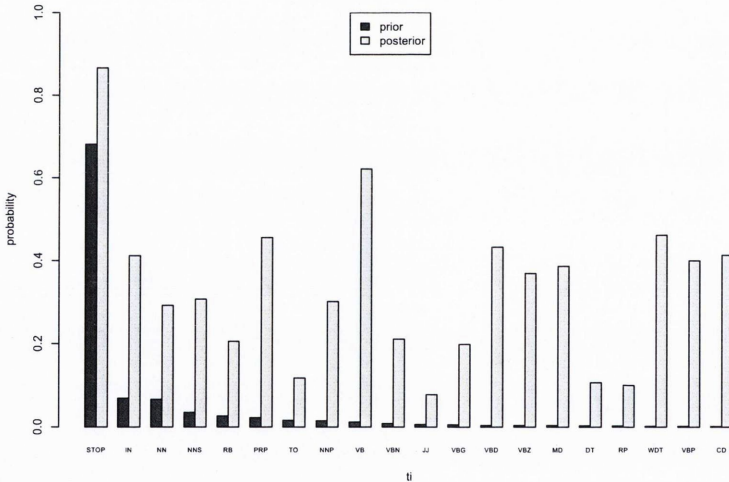


Figure 5.5: Prior and posterior (positive adaption) probabilities for modifier POS labels at conjunct depth 1

For modifier nodes we also found similar evidence of symmetry across conjuncts, both for non-terminal labels and head part-of-speech tags. Figure 5.5 displays results for the top twenty modifier part-of-speech tags across conjuncts at depth 1.



Depth	HeadEvents	%M $C_h$ L-R	%M $C_h$ H-F
1	14,156 (14%)	81	87
2	20,932 (21%)	37	47
3	16,671 (17%)	19	24
4	12,689 (13%)	10	13
5	9559 (9.7%)	5.6	7.1
6	7316 (7.4%)	2.9	3.7
7	5319 (5.4%)	1.0	1.4
8	3882 (3.9%)	0.54	0.57
9	2705 (2.7%)	0.26	0.26
10	1830 (1.9%)	0.11	0.16

Table 5.3: Percentage Match(%M) of head event labels  $C_h$  in right-of-head conjuncts with the corresponding label in the head conjunct, grouped by Depth. Percentage match for head conjunct nodes collected in both a left-to-right (L-R) traversal and head-first (H-F) traversal are shown.

Finally, in an effort to summarise results across different conjunct depths, we show the percentage of times the POS tag and non-terminal label of a node,  $N_i$  in the post-CC conjunct matches the POS tag and non-terminal label of the corresponding node  $N_{i_{preCC}}$  in the head conjunct. When counting matches, if there is no corresponding node in the head conjunct for a node in the post-CC conjunct then this counts as a non-match.

Table 5.4 shows the percentage of POS tags ( $t_i$ ) and non-terminal labels ( $C_i$ ) of modifier nodes in post-CC conjuncts that have the same value for POS and non-terminal labels as  $N_{i_{preCC}}$  in the head conjunct. Results are shown for both head-first and left-to-right traversals for each level in the coordinate phrases. For depth 0 and depth 1 non-terminal labels across conjuncts match more often than part-of-speech tags. From depth 2 on, the percentage of matches for non-terminal labels and POS tags is similar. This is probably because the conjunct nodes from depth 2 and deeper are more likely to be pre-terminal nodes, where the node label and POS tag are the same for a given node. For both POS tags and non-terminal labels, there is little difference whether the pre-CC nodes are retrieved head-first or left-to-right.

Table 5.3 shows the percentage of matches of the head non-terminal label in the post-CC conjunct with the corresponding non-terminal label in the pre-CC conjunct for both left-to-right and head-first traversals. Head-first traversals of the pre-CC

Depth	$ DepEvents $	%M $C_i$ L-R	%M $C_i$ H-F	%M $t_i$ L-R	%M $t_i$ H-F
0	15,840 (5.4%)	93	93	65	65
1	40,429 (14%)	69	68	64	63
2	59,907 (21%)	49	49	48	47
3	45,623 (16%)	31	30	31	30
4	35,511 (12%)	17	16	16	16
5	26,538 (9.1%)	8.5	8.4	8.4	8.2
6	20,423 (7.0%)	4.1	4.3	4.1	4.3
7	14,681 (5.0%)	1.8	1.8	1.7	1.8
8	10,771 (3.7%)	0.95	0.90	0.95	0.88
9	7472 (2.6%)	0.37	0.40	0.37	0.40
10	5039 (1.7%)	0.28	0.26	0.28	0.26
11	3388 (1.2%)	0.059		0.059	

Table 5.4: Percentage Match(%M) of  $C_i$  and  $t_i$  labels of dependent events in right-of-head conjuncts with the head conjunct, grouped by depth. Percentage match for head conjunct nodes collected in both a left-to-right (L-R) traversal and head-first (H-F) traversal are shown. The total number of dependent events ( $|DepEvents|$ ) in post-CC conjuncts for each level is displayed.

conjunct return more matching labels. Comparing the data in Table 5.4 and Table 5.3, the percentage of label matches is greater for head labels than for modifier labels at depth 1 but not for depths greater than one. However, if we remove all STOP events<sup>4</sup> from the dependent events set and then compare the percentage of dependent node label matches with the percentage of head node label matches, the percentage of head node matches is greater at all depths.

There is clearly evidence of a bias towards symmetry in the syntactic structure of conjuncts, although this symmetry diminishes the deeper the level in the conjuncts. Learning a bias toward parallelism should improve the parsing model's ability to correctly attach the coordination conjunction and second conjunct to the correct position in the tree. In Figure 5.1 for example, a preference for symmetry in conjuncts might help the model to attach the CC node and  $NP[states]$  subtree to the  $NP[plains]$  node due to the fact that the two  $NPs$  have almost identical internal structure.

---

<sup>4</sup>i.e. where the non-terminal label in the post-CC conjunct is the STOP symbol.

### 5.3 Modelling Symmetry in Conjuncts

In the Collins generative history-based model a tree is generated top-down head-first and features are limited to being functions of the tree generated so far. Thus the task is to incorporate a feature into the model that captures a particular bias yet still adheres to these derivation-based restrictions. Each node in the tree in Figure 5.1 is annotated with the order in which the nodes are generated (we omit, for the sake of clarity, the generation of the *STOP* nodes). Note that when the decision to attach the second conjunct to the head conjunct is being made (i.e. Step 11, when the *CC* and *NP[states]* nodes are being generated) the internal structure of the sub-tree rooted at *NP[states]* has not yet been generated. Thus at the point that the conjunct attachment decision is made it is not possible to use information about symmetry of conjunct structure as we do not know yet what the structure of the second conjunct will be.

It is possible, however, when generating the internal structure of the second conjunct to condition on structure of the already generated head conjunct. In order to allow the model to learn a preference for symmetric structure, we introduce new conditioning features: when the structure of the second conjunct is being generated we condition on features which are functions of the first conjunct, returning for example the part-of-speech tag of  $N_{i_{preCC}}$  as a feature when predicting a POS tag for a node  $N_i$  in the post-CC conjunct.

The usual parameter classes for estimating the probability of the head label,  $C_h$ , and the part-of-speech label of a modifier node,  $t_i$ , are (as outlined also in §3.5.2):

$$P_{C_h}(C_h|C_p, w_p, t_p, t_{gp}) \quad (5.1)$$

$$P_{t_{left}}(t_i|dir, C_p, C_h, w_p, dist, t_{i-1}, t_{i-2}, C_{gp}) \quad (5.2)$$

$$P_{t_{right}}(t_i|dir, C_p, C_h, w_p, t_p, dist, t_{i-1}, t_{i-2}, C_{gp}) \quad (5.3)$$

Instead of the above parameter classes we created two new parameter classes which are used only in the generation of post-CC conjunct nodes. These parameter classes are as follows:

$$P_{C_h,conjunct}(C_h|\gamma(headConjunct), C_p, w_p, t_p, t_{gp}, depth) \quad (5.4)$$

$$P_{t_i,conjunct}(t_i|\alpha(headConjunct), dir, C_p, w_p, t_p, dist, t_{i,1}, t_{i,2}, depth) \quad (5.5)$$

where  $\gamma(headConjunct)$  returns the non-terminal label of  $N_{i_{preCC}}$  for a head node,  $N_i$ , and  $\alpha(headConjunct)$  returns the POS tag of  $N_{i_{preCC}}$  for modifier node,  $N_i$ . Both functions return *+NOMATCH+* if there is no  $N_{i_{preCC}}$  for the node being generated. *depth* is the level of the post-CC conjunct node  $N_i$ . The parameter class (5.4) replaces that of (5.1) in the generation of post-CC conjunct nodes and the parameter class of (5.5) is used in the generation of both left and right modifier nodes (replacing both (5.2) and (5.3)).

## 5.4 Relation to Previous Work

Dubey et al. [2005] demonstrate the prevalence of parallel structures across conjuncts in coordinate NP data from the Penn Treebank. Drawing data from all  $NP_1$  *CC*  $NP_2$  constructions, they focus on five types of syntactic construction (for example the construction:  $NP \rightarrow DT JJ NN$ ) and measure frequencies of occurrence of the syntactic constructions in  $NP_1$  and  $NP_2$ . They compare the prior probability of a particular construction occurring in  $NP_2$ , with the probability of the construction occurring in  $NP_2$ , given it has occurred in  $NP_1$ . This latter probability they call *positive adaption* after the work on lexical priming of Church [2000]. They find that, for all but one of the construction types examined, a given construction is more likely to occur in  $NP_2$  given it has occurred in  $NP_1$ . Interestingly, the only construction type where the prior probability was higher than the positive adaption probability was the case of the type:  $NP \rightarrow NN$ . We would guess that the reason for this is because a coordinate NP structure such as  $(NP (NP (NN)) CC (NP (NN)))$  would, in fact, be inconsistent with the Penn guidelines (the correct structure being  $(NP (NN CC NN))$ ) and, therefore, although it does occur in the data, it would not do so as often as phrases which are consistent with the guidelines.

In Section 5.2 we measure symmetry of conjunct structure in our training data by counting how often the non-terminal label in a post-CC conjunct of a coordinate phrase



matches the corresponding non-terminal label in the pre-CC conjunct. Unlike Dubey et al. [2005] we do not focus on NPs alone but instead look at symmetry in conjunct structure across all conjunct types, with the exception of flat NPB constructions. In addition where Dubey et al. [2005] measured symmetry at depth 1 only of the conjuncts, we look at the parallelism effect for different conjunct depths. A final difference is that rather than comparing sequences of non-terminals we compare individual nodes.

In terms of coordination disambiguation, several previous attempts have attempted to take advantage of the tendency for parallel structures across conjuncts, as described in §2.6. Insofar as we do not separate coordination disambiguation from the overall parsing task, our approach resembles the efforts to improve a coordination disambiguation in the discriminative rerankers of Charniak and Johnson [2005] and Ratnaparkhi et al. [1994], where both rerankers include features to capture syntactic parallelism across conjuncts at various depths.

## 5.5 Summary

We have demonstrated that a significant level of parallelism exists in the syntactic structure of conjuncts in the WSJ. The symmetric effect holds true both for non-terminal labels and, to a lesser extent, for part-of-speech labels and is evident at increasing conjunct depths, though unsurprisingly parallelism decreases with increasing depth. The mechanism for aligning nodes in post-CC conjuncts with nodes in pre-CC conjuncts can be incorporated into the probability model in order to encourage the model to give more weight to syntactic structures which exhibit parallelism. This is done in the following manner: when generating syntactic structure in a post-CC conjunct, the model conditions on aligned structure in the pre-CC conjunct. In Chapter 7 we give details of experiments with the parameter classes introduced in this chapter and show the results of these changes to the baseline model.

## Chapter 6

# Noun Phrase Coordination Error Analysis

### 6.1 Introduction

In this chapter we look at two different causes for the incorrect bracketing of coordinate noun phrases in the model described in Chapter 3. Section 6.2 examines inconsistencies in the annotation of coordinate NPs in the Penn Treebank which can lead to errors in coordination disambiguation. We show how some of the types of coordinate noun phrase inconsistencies can be automatically detected.

In Section 6.3 we describe a method of tree alignment to aid error analysis. We also discuss how the different head-finding rules for coordinate noun phrases and coordinate base noun phrases can negatively affect coordination disambiguation. Section 6.3.1 suggests a minor modification to the head-finding rules for base noun phrases so that the lexical item chosen to head the entire phrase more closely resembles the head chosen for other types of coordinate noun phrase.

## 6.2 Bracketing Guidelines for the Penn Treebank and Inconsistencies in WSJ Coordinate Noun Phrase Annotation

The annotation of noun phrases in the Penn Treebank [Bies et al., 1995] follows somewhat different guidelines to that of other syntactic categories. Because the interpretation of nominal modifiers is highly ambiguous and often subject to individual interpretation, no internal structure is shown for nominal modifiers. Hence the following flat structures (examples taken from [Bies et al., 1995]): (*NP the primary college entrance examination*) and (*NP U.S. patent and copyright owners*). For noun phrases with more than one head noun, if the only unshared modifiers in the constituent are nominal modifiers, then a flat structure is also given. Thus in (*NP the Manhattan phone book and tour guide*) a flat structure is given because although *the* is a non-nominal modifier it is shared, modifying both *tour guide* and *phone book*, and all other modifiers in the phrase are nominal. Note that even though *phone* is clearly unshared in that it modifies *book* but not *tour guide*, no internal structure is shown because it is a nominal premodifier.

However, it happens relatively often in the WSJ Treebank that these guidelines are not followed, and coordinate noun phrases which should be annotated flat are instead given internal structure. Take the following example (this time we show the POS tags as well) extracted from the treebank:

(a) (*NP (NP (NNS controllers))(CC and)(NP (NN disk)(NNS drives))*)

According to the guidelines, the phrase should be bracketed flat. Out of 1,417 examples of noun phrase coordination in sections 02 to 21 inclusive, involving phrases containing only nouns (common nouns or a mixture of common and proper nouns) and the coordinating conjunction, we found 21.3%, contrary to the guidelines, were given internal structure. When all proper nouns are involved it is even more common to encounter a coordinate NP showing internal structure where officially they should be given a flat structure, for example:

(b) (*NP (NP (NNP Rainman))(CC and)(NP (NNP Batman))*)

In the guidelines, however, it is recognised that proper names are frequently annotated

with internal structure. We found 1,369 examples of coordinate noun phrases where all nouns were proper. Of these 29.4% were given structure.

Another common source of inconsistency in coordinate noun phrase bracketing occurs when a non-nominal modifier appears in the coordinate noun phrase. As previously discussed, according to the guidelines the modifier is annotated flat if it is shared. Where it is unclear if a non-nominal modifier is shared or not, the default is to bracket as shared. When the non-nominal modifier is unshared, more internal structure is shown:

(c) *(NP (NP (NNS fangs)) (CC and) (NP (JJ pointed) (NNS ears)))*

(d) *(NP (NP (DT the) (NNP U.S.)) (CC and) (NP (NNP Europe)))*

We found on inspection that sometimes a flat annotation was given, when in fact more structure should have been shown. Take the following two examples extracted from the treebank, which should in fact be given the more structured bracketing shown in Figure 6.1 Tree 1 and 2 respectively:

(e) *(NPB (NN oversight)(CC and)(JJ disciplinary)(NNS procedures))*

(f) *(NPB (JJ moderate)(CC and)(JJ low-cost)(NN housing))*

Following the guidelines any coordinate base noun phrase which ends with the following tag sequence can be automatically detected as incorrectly bracketed: *CC/non-nominal modifier/noun*. This is because either the non-nominal modifier, which is unambiguously unshared, is part of a noun phrase as in Figure 6.1, Tree 1 or it conjoined with another modifier as in Figure 6.1, Tree 2.<sup>1</sup> We found 202 examples of this in the training set, out of a total of 4,895 coordinate base noun phrases.

Finally, inconsistencies in POS tagging can also lead to problems with coordination. Take the bigram *executive officer*. We found 151 examples in the training set of a base noun phrase which ended with this bigram. 48% of the cases were POS tagged *JJ NN*,

---

<sup>1</sup>Note however that *CC/non-nominal modifier/noun/noun* can not be automatically classified as inconsistent with the Treebank guidelines. For example, *(NPB (JJ personal) (NN computer) (CC and) (JJ electronic) (NN equipment) (NN maker))* is correctly bracketed flat because 'No internal structure is shown for conjoined nominal premodifiers...Even in the case where a nominal premodifier is adjectively modified, the entire structure is left flat'.



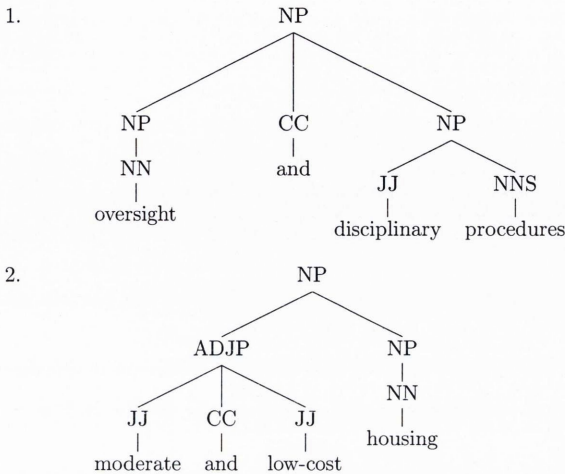


Figure 6.1: Correct Parse tree bracketing according to the Penn Guidelines

52% tagged *NN NN*.<sup>2</sup> This has repercussions for coordinate noun phrase structure, as the presence of an adjectival pre-modifier indicates a structured annotation should be given.

These inconsistencies pose problems both for training and testing. With a relatively large amount of noise in the training set the model learns to give too high a probability to structures which should be very unlikely. In testing, given inconsistencies in the gold standard trees, it becomes more difficult to judge how well the model is doing.

### 6.3 NPB Head-Finding Rules

In order to gain more insight into the type of errors being made in coordinate structures we compared the erroneous coordinate phrases proposed by our memory-based baseline model on the validation set, with the corresponding oracle coordinate phrases, where the oracle subtrees are correct. For each correct NP coordinate phrase in the oracle

<sup>2</sup>According to the POS bracketing guidelines [Santorini, 1991] the correct sequence of POS tags should be *NN NN*.

trees that did not exist in the trees selected by the baseline model, we retrieved the (incorrect) sub-tree from the baseline model set that contained the same coordinating conjunction. We then aligned the two subtrees so that they spanned the same number of words. We did not align trees where the oracle and baseline model tree contained crossing brackets.<sup>3</sup> Figure 6.2 demonstrates how trees were aligned. Tree 1 contains the correct coordination dependencies which occurred in the oracle tree. Tree 2 shows the corresponding coordinate noun phrase returned by the baseline model. Tree 3 shows the oracle tree subtree after the oracle and baseline subtrees have been aligned.

Out of a total of 190 coordinate noun phrases, including base noun phrases, where the oracle subtree was correct and the baseline model subtree incorrect, 156 trees were aligned in this manner. This left us with a set of 156 paired coordinate noun phrases, where each pair contained the incorrect structure chosen by our model as well as the correct version of the subtree.

Aligning trees allowed us to easily examine the types of error being made in coordinate structures. We could also compare the probability estimates the generative reranker gives for the two trees. We found that for 25% of the pairs, the correct coordinate sub-tree was correctly assigned a higher probability, this despite the fact that this structure was not the structure that ends up in the highest scoring parse according to our model. One reason this might occur could be to do with factors unrelated to the coordinate noun phrase in question but instead related to the probability given to other structures in the tree of which the coordinate NP is but a component. This is discussed further in §7.4. Another reason this phenomenon occurred was because the probabilities we compared for the two subtrees do not take into account the probabilities of the head word (and head POS tag, non-terminal label etc.) of the subtrees, given their previously generated parse structures. These missing generative terms can have an important effect, particularly when the heads of the two subtrees are different, which was the case to a significant extent in the aligned trees. In such cases, the probabilities of the subtrees are to some extent incomparable.

The choice of head affects the various dependencies in the model. Head-finding rules for coordinate NPBs differ from coordinate NPs.<sup>4</sup> Take the following two versions of

---

<sup>3</sup>Two trees contain crossing brackets if the constituents in one tree cross over constituent boundaries in the other tree. See [Manning and Schütze, 2001, p. 434] for an explanation, with illustration, of crossing brackets.

<sup>4</sup>The head rules used in the baseline model can be found in the Appendix of [Collins, 1999].

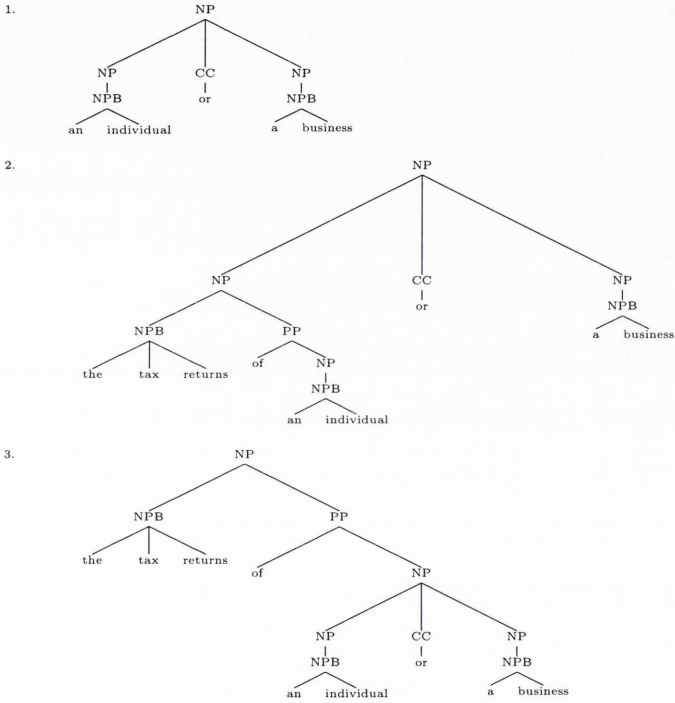


Figure 6.2: Tree 1: The correct oracle coordinate NP. Tree 2: The incorrect coordinate NP returned by the baseline model. Tree 3: The oracle tree aligned with Tree 2.

the noun phrase *hard work and harmony*:

(g) (NP (NPB *hard work and harmony*))

(h) (NP (NP (NPB *hard work*)) and (NP (NPB *harmony*))).

In the first example, *harmony* is chosen as head word of the NP; in example (h) the head of the entire NP is *work*. In the case of two coordinate NPs which, as in the above example, cover the same span of words and differ only in whether the coordinate noun phrase is flat as in (g) or structured as in (h), the choice of head for the phrase is not particularly informative. In both cases the head words being coordinated are the same and either word could plausibly head the phrase; discrimination between trees in such cases should not be influenced by the choice of head, but rather by other, salient features that distinguish the trees.<sup>5</sup>

### 6.3.1 Modifying the NPB Head-Finding Rules

In order to avoid discrimination based on the differing head-finding rules of coordinate NPs and NPBs, we would like to alter the head-finding rules for coordinate NPBs so that the word chosen to head the entire coordinate noun phrase would more often coincide with that chosen in non-base noun phrases. One of the difficulties lies in detecting which are the two nouns being coordinated. A rule that, for example, would always choose the noun to the left of the CC to head the base noun phrase, risks choosing an obviously incorrect head in a phrase such as *French and German cars*, where the nominal modifiers are coordinated and the head of the noun phrase is clearly *cars*.

[Collins, 1999, p. 238] gives the head-finding rules used in the Collins parser. As there are no rules explicitly for base noun phrases we can assume they are the same as the rules for noun phrases. Coordinated phrases have their own special treatment with regard to head-finding. NPBs are not treated as coordinated phrase in the Collins model (for reasons discussed in §1.3). In coordinate phrases the head always comes *before* the CC node, but this is not the case for NPBs. Generally in NPBs the right-most noun is chosen to head the phrase. For example, in a base noun phrase such

<sup>5</sup>For example, it would be better if discrimination was largely based on whether *hard* modifies both *work* and *harmony* (g), or whether it modifies *work* alone (h).



as  $NPB \rightarrow noun_i CC noun_j noun_k$ , the head of the phrase is (usually) the rightmost noun in the phrase.

In our model, in a base noun phrase with a sequence of children:<sup>6</sup>

$noun_i CC noun_j noun_k$

the head rules remain unchanged and the head of the phrase is, as before, (usually) the rightmost noun in the phrase. That is, when the pattern  $noun_i CC noun_j noun_k$  occurs - when  $noun_j$  is immediately followed by another noun - the default is to assume nominal modifier coordination and the head rules stay the same. In such cases the bilinear coordinate head-head dependencies of §4.3.2 are modelled as  $P_{coordWord}(noun_i|noun_j)$ .

The slight modification to the head rules for NPBs that we make is as follows: when  $noun_j$  is *not* immediately followed by a noun, in any NPB containing the pattern  $noun_i CC noun_j$ , then the noun chosen to head the entire phrase is the noun preceding the CC:  $noun_i$ . (This alteration to the original head-finding rules is not implemented if there is more than one CC node in the NPB.) The head-head dependencies in such cases are then modelled as  $P_{coordWord}(noun_j|noun_i)$ .

In addition, if the following pattern occurs:  $NPB \rightarrow C_i CC C_j$ , where the label  $C_i$  is the same as the label  $C_j$ , and both nodes are pre-terminals, then, in the new head-finding rules, the head of the phrase is the node labelled  $C_i$ . Note that there is no requirement that the labels be nouns.

These two modifications to the head-finding rules are both aimed at making the rules for coordinate NPBs more similar to those for coordinate NPs for the reasons outlined in §6.3. We do not, however, suggest that the head rule modifications proposed here offer a complete solution. In a phrase such as *dolls and toy cars*, for example, *dolls* would be the head chosen if the phrase were (incorrectly according to the Penn guidelines) given internal structure, as in  $(NP (NP (NPB dolls)) and (NP (NPB toy cars)))$ . On the other hand, even with the modified rules, in the tree  $(NPB dolls and toy cars)$ , *cars* would be the head word. Altering the head rules also had side-effects which necessitated new features for the generation of modifier nodes with NPB parents, which we discuss in more detail in §7.3.3.

---

<sup>6</sup>Note there can be other children to the left or right of this sequence.

## 6.4 Relation to Previous Work

Heuristics to detect particular inconsistencies in treebank annotation have been explored before, for example, when deriving a categorical grammar style annotation from Penn Treebank trees in [Hockenmaier and Steedman, 2005]. Blaheta [2002] characterises types of errors in inconsistency in corpus annotation and also gives examples of inconsistency detection using rules which search for specific inconsistencies. More general approaches to treebank inconsistency detection, which do not rely on hand-crafted heuristics, are outlined in Dickinson and Meurers [2005]. In testing their method they take a similar approach to us insofar as they eliminate the noisy events. Their method was tested on the WSJ corpus by retraining a PCFG parser on the training data after having eliminated all rules in the training data that arose from local trees considered to be errors by the method. They then compared the result with the parser trained on the original training set. The result on an (unchanged) section 23 showed small yet significant increases in precision and recall.

## 6.5 Summary

We have shown that coordinate noun phrase data appears to be particularly noisy in the WSJ Penn Treebank. Inconsistencies in coordinate noun phrase data make it harder for a model to learn the correct bracketing for coordinate NPs. We demonstrated that for some of these inconsistencies it is possible to automatically detect when an NP tree in the treebank is not bracketed according to the Penn treebank bracketing guidelines. In chapter 7 we will use this automatic detection of inconsistent trees to clean noisy data in the training and test sets.

In this chapter we also described a useful method to facilitate error analysis and discussed how differing head-finding rules for NPs and NPBs can negatively affect coordination disambiguation. We suggest a minor alteration in the head-finding rules for coordinate base noun phrases so that the lexicalisation process for coordinate NPBs is more similar to that of other NPs. In Chapter 7 we demonstrate the effects of changing the NPB head-finding rules on the accuracy of the model.

# Chapter 7

## Experimental Evaluation - Coordination

### 7.1 Introduction

This chapter gives the details of our experiments on improving NP coordination disambiguation, implementing the ideas of the previous chapters. We begin by outlining how we select our coordination test and validation sets and under what criteria a coordination dependency is taken to be correct. We then step through each set of experiments, from those involving eliminating noisy data from the training set, to the introduction of parameter classes that capture symmetry in conjunct structure, to changes in the NPB head-finding rules, to the experiments on modelling conjoined head words. We show the effect of each experiment on the validation set, the overall effect on the test set, and conclude with a discussion of the results achieved.

### 7.2 Experimental Evaluation

For our experiments on coordination disambiguation our baseline model is that described in Chapter 3, where  $k$ -NN is used for parameter estimation. As in Chapter 3, the experiments outlined in this chapter take place in the context of parse reranking, where the  $n$ -best output from Bikel's parser is reranked according to our parsing model. Sections 02 to 21 are again used for training. The coordination test set is taken from

section 23 and the coordination validation set taken from the remaining WSJ sections. Only sentences containing 40 words or less were used for testing and validation.

As outlined in §6.2 the Penn Treebank guidelines are somewhat ambiguous as to the appropriate bracketing for coordinate noun phrases which consist entirely of proper nouns. We therefore do not include, in the coordination test and validation sets, coordinate noun phrases where in the gold standard NP the leaf nodes consist entirely of proper nouns (or CCs or commas). In doing so we hope to avoid a situation whereby the success of the model is measured in part by how well it can predict the often inconsistent bracketing decisions made for a particular portion of the treebank.

In addition, and for the same reasons, a tree is not included when calculating coordination precision and recall of the model if the gold standard tree is inconsistent with the guidelines in either of the following two ways: the gold tree is a noun phrase which ends with the sequence *CC/non-nominal modifier/noun*; the gold tree is a structured coordinate noun phrase where each word in the noun phrase is a noun (recall from §6.2 that for this latter case the noun phrase should be flat - an NPB - rather than a noun phrase with internal structure). Call these inconsistencies type *a* and type *b* respectively.

In total, 296 coordination dependencies were excluded in this manner from the validation set and 134 coordination dependencies excluded from section 23. This left us with a coordination validation set consisting of 1064 coordinate noun phrase (including base noun phrase) dependencies and a test set of 416 coordinate NP/NPB dependencies from section 23.

A coordinate noun phrase dependency was deemed correct if the parent constituent label, and the two conjunct node labels (at level 0) match those in the gold subtree and if, in addition, each of the conjunct head words are the same in both test and gold tree. This follows the definition of a coordinate dependency in [Collins, 1999]. As in [Collins, 1999] all labels which are part-of-speech tags are relabelled TAG in order to avoid errors in tagging being counted as dependency errors. In our tests, for NPBs the first conjunct was taken to be the head node of the phrase, using the original head rules. The second conjunct was taken as the node generated directly after the coordinating conjunction. Based on this criteria, the baseline *f*-scores for test and validation set were 69.1% and 67.1% (see Table 7.1) respectively. The coordination *f*-score for the oracle trees on section 23 is 83.56%. In other words: if an 'oracle' were to choose



Model	$f$ -score	significance
1. Baseline	67.1	
2. NoiseElimination	68.7	$\gg 1$
3. Symmetry	69.9	$> 2, \gg 1$
4. NPB head rule	70.6	NOT $> 3, > 2, \gg 1$
5. $P_{coordWord}$ WSJ	71.7	NOT $> 4, > 3, \gg 2$
6. BNC data	72.1	NOT $> 5, > 4, \gg 3$
7. $sim(w_i, w_p)$	72.4	NOT $> 6, \text{NOT } > 5, \gg 4$

Table 7.1: Results on the Validation Set. 1064 coordinate noun phrases dependencies. In the significance column  $>$  means at level .05 and  $\gg$  means at level .005, for McNemar’s test of significance. Results are cumulative.

from each set of  $n$ -best trees the tree that maximised constituent precision and recall, then the resulting set of oracle trees would have a NP coordination dependency  $f$ -score of 83.56%. For the validation set the oracle trees coordination dependency  $f$ -score is 82.47%. The labelled precision and recall scores for the oracle and baseline trees for section 23 are displayed in Table 7.2.

## 7.3 Experimental Details and Results

In this section we give a breakdown of results on the validation set (see Table 7.1), as well as the overall results of all experiments on the coordination dependency  $f$ -score of section 23 (see Table 7.2). Results reported on the validation set are cumulative. All statistical significance tests were carried out using McNemar’s test [Dietterich, 1998] for significance, based on the number of correct/incorrect coordination dependencies in the data sets.

### 7.3.1 Eliminating Noisy Data

Our first experiments consisted of attempts to reduce noise in the training data. We did this by automatically detecting type  $a$  and type  $b$  inconsistencies (defined in Section 7.2) and eliminating them from the training set. The effect of this on the validation set is outlined in Table 7.1 (row 2). Eliminating the noisy data resulted in a statistically significant ( $p < 0.005$ ) improvement in coordination accuracy, with the  $f$ -score rising

from the baseline 67.1% to 68.7%.

### 7.3.2 Modelling Symmetry in Conjunct Structure

Our next changes to the baseline model involved implementing the parameter classes described in §5.3, which aim at introducing a bias toward parallelism in conjunct structure. For the parameter class of (5.4) in §5.3 which models the probability of  $C_h$  (the head child non-terminal label) in a post-CC conjunct given its history, the best results occurred when the parameter class was used only at depths 1 and 2 of the conjuncts, although the training examples for this parameter class contained head events from all post-CC conjunct depths. The parameter class of (5.5) in §5.3 was used for predicting POS tags at level 1 in post-CC conjuncts, although again the training set contained events from all depths. We did not restrict use of these parameters to noun phrases conjuncts only but used the parameter class for all types of conjunct. The result of these new parameter classes was a rise in  $f$ -score accuracy to 69.9%, a significant ( $p < 0.05$ ) rise in coordination accuracy (Table 7.1 row 3) from the previous score of 68.7%.

### 7.3.3 NPB Head-Finding Rule and New Features for NPBs

As suggested in §6.3.1 we altered the head-finding rules for base noun phrases. At this point we also introduced two new types of conditioning features to the history of parameter class  $P(C_i, t_i | C_p = NPB, H(i))$ . In the memory-based model presented in Chapter 3, three conditioning features for this parameter class are  $C_{i-1}$ ,  $C_{i-2}$ , and  $C_{i-3}$  (the non-terminal labels of the three previously generated nodes). Instead, we found it useful to chunk the three previously generated non-terminal labels together into one feature. The idea behind this feature was to make certain sequences, like: *DT JJ NN CC*, more unlikely. Initially, after altering the head-finding rules, we found problems were being caused because generating a coordinating conjunction and subsequent nodes to the right of the head word was too unlikely. We found adding a new ‘comma’ distance feature for the generation of nodes to the right of the head node helped. We added a boolean feature which returns true if a punctuation node (with POS tag ‘,’ or ‘:’) has already been generated as a sibling to the node in question, false otherwise. The effect of these changes are displayed in Table 7.1 (row 4). There was a rise in  $f$ -score from

the previous result of 69.9% to 70.6%, though the change in coordination accuracy was not statistically significant.

### 7.3.4 Modelling Conjoined Head Nouns

We now turn to the experimental details for the word graph described in §4.3.2. For building the word graph we extracted 9961 coordinate noun pairs from the WSJ training set and 815,323 pairs<sup>1</sup> from the BNC. As links between pairs are symmetric this resulted in a total of 1,650,568 coordinate noun events stored in the graph. All words were collapsed to lower case, and every digit replaced by the special character  $\mathcal{E}$ . The final graph consisted of 82,579 nodes, or word types.

This word graph was then used for the estimation of the parameters of the  $P_{coordWord}$  parameter class introduced in §4.3. For all our experiments with  $P_{coordWord}$  the parameter class is used both for NPs and NPBs. In our first experiments we estimated  $P_{coordWord}$  from the graph without using the similarity function and used only two layers of back-off combined using Witten-Bell interpolation, as in:

$$P_{coordWord}(n_i|n_j) = \lambda_{n_j} P_{WG}(n_i|n_j) + (1 - \lambda_{n_j}) P_{kNN}(n_i|H(i)) \quad (7.1)$$

where  $P_{WG}(n_i|n_j)$  is the maximum likelihood estimate calculated from the events stored in the graph and in  $P_{kNN}(n_i|H(i))$  the history is simply the POS tag of  $n_i$ . For the  $P_{kNN}(n_i|H(i))$  estimate, as with word estimates in the original baseline model, words occurring less than 5 times were mapped to the  $+UNKNOWN+$  token. We found that in the calculation of  $\lambda_{n_j}$ , above, the best results occurred when the constant  $C$  was set to 5, the same setting as for the Witten Bell estimations in the baseline model. Calculating  $P_{coordWord}$  in such a fashion, Table 7.1 shows the effect of this parameter class estimated from a word graph that contained WSJ data only (row 5), and then from the word graph with the addition of the BNC data (row 6).

We incorporated the similarity measure (introduced first in §4.2.1) into the estimate of  $P_{coordWord}$  in the manner described in §4.3.2, and repeated here for convenience:

$$P_{coordWord}(n_i|n_j) = \lambda_{n_j} P_{WG}(n_i|n_j) + (1 - \lambda_{n_j}) P_{simInterp}(n_i|n_j) \quad (7.2)$$

---

<sup>1</sup>Note that some of these pairs of nouns from the BNC were extracted from lists in the manner described in §4.3.2.

Model	NPccPrecision	NPccRecall
Baseline	66.03	74.29
FinalModel	70.46	77.40
Oracle	79.57	87.98

Table 7.2: Results for Section 23. 416 coordinate noun phrase dependencies

where

$$P_{simInterp}(n_i|n_j) = \lambda'_{n_j} P_{sim}(n_i|n_j) + (1 - \lambda'_{n_j}) P_{kNN}(n_i|H(i)) \quad (7.3)$$

In practice it proved too computationally expensive to calculate similarity measures for every vertex in the graph. For the estimation of  $P_{sim}(n_i|n_j)$  we found the best results were obtained when the neighbourhood of  $n_j$  was taken to be the  $k$ -nearest neighbours of  $n_j$  from among the nodes directly connected to  $n_j$ , where  $k$  is 1000. For the calculation of  $\lambda_{n_j}$  in (7.2) we set the constant  $C$  to 10, and for the calculation of  $\lambda'_{n_j}$  in (7.3) it was set to 5. As before, for the  $P_{kNN}(n_i|H(i))$  estimate the history was simply taken to be the noun  $n_i$ 's POS tag. Table 7.1 shows the effect of the  $P_{coordWord}$  parameter class estimated with the word similarity measure (row 7).

Though none of the three individual changes to model discussed in this subsection (first using  $P_{coordWord}$  on WSJ alone, then on BNC data and finally with the similarity metric) resulted in statistically significant changes in coordination accuracy, taken together the changes result in a rise of  $f$ -score accuracy from 70.6% to 72.4%, a statistically significant result ( $p < 0.005$ ).

### 7.3.5 Results

The overall result on the test set of all these experiments was an increase in coordinate noun phrase  $f$ -score from 69.91% to 73.77%. This represents a 13% relative reduction in coordinate  $f$ -score error over the baseline, and, using McNemar's test [Dietterich, 1998] for significance, is significant at the 0.05 level ( $p = 0.034$ ). The reranker  $f$ -score for all constituents for section 23 rose slightly to 89.6%.<sup>2</sup>

Finally, for the sake of completeness, we report results on an unaltered coordination

---

<sup>2</sup>This is the evalb score on the full trees of section 23, not excluding any coordinate NPs.



test set, that is, a test set from which no noisy events were eliminated. The baseline coordination dependency  $f$ -score for all NP coordination dependencies (550 dependencies) from section 23 is 69.27%. This rises to 72.74% when all experiments described in Section 7.3 are applied, which is also a statistically significant increase ( $p = 0.042$ ).

### Comparing Coordination Results with Previous Work on Coordination Disambiguation

Though the work of [Resnik, 1999, Goldberg, 1999, Nakov and Hearst, 2005] on coordination disambiguation is also tested on WSJ data, it is nevertheless not possible to compare our coordination disambiguation results with the results of these other systems. As discussed in more detail in §2.6, the approaches of both [Resnik, 1999] and [Nakov and Hearst, 2005] aim to show more structure than is shown in trees following the Penn guidelines, whereas in our approach we aim to reproduce Penn guideline trees. The learning task is therefore different. In the probabilistic approach to coordination disambiguation of Goldberg [1999] the system is tested on a particular type of coordination construction involving prepositional phrases. While this type of construction does form a proportion of our test set it is nevertheless difficult to compare results. Firstly, Goldberg [1999]’s system is not tested on a test corpus. In addition, results are given only for a development set of 500 phrases extracted from the annotated treebank and with no details given on which WSJ sections the examples were extracted from.

Finally, again as discussed in §2.6, the discriminative reranker of Charniak and Johnson [2005] contains coordination specific features and is tested on Section 23 of the Penn Treebank. However the effect on coordination disambiguation is not tested, only the labelled precision and recall  $f$ -score results for all constituents are given. As both the [Charniak, 2000] base parser and the [Charniak and Johnson, 2005] reranker are widely available we carried out an experiment to compare the NP coordination dependency score of the base parser and the discriminative reranker. Taking the output of the Charniak base parser and the Charniak and Johnson reranker on section 23, we calculated the NP coordination dependency  $f$ -score in the fashion described in §7.2, for the same test set of 416 coordinate NPs. The base parser achieves a coordination dependency  $f$ -score of 72.05%, which increased to 80.22% for the reranking parser. Although both results are impressive they are not directly comparable to the

results reported for our baseline model and reranker. Firstly, the Charniak base parser is not identical to the Collins parser and achieves a considerably higher labelled precision/recall  $f$ -score to both the Collins parser and the model described in Chapter 3 of this thesis. Thus one would expect the baseline coordination results to be higher for the Charniak parser. For the reranking experiments, the Charniak parser also produces higher quality  $n$ -best lists than the Collins parser (and than Bikel's emulation of the Collins parser)<sup>3</sup> which tends to lead to higher reranker scores (see discussion § 2.3.1). Finally, the reranking parser is discriminative and includes some 1,148,697 features, of which only 32 are coordination-specific features. Thus, many features contribute to the selection of the best tree from an  $n$ -best list and it is not possible to say to what extent the high coordination dependency score for the reranking parser is due to the coordination-specific features or due to other factors which contribute to picking overall high-scoring trees.

## 7.4 Discussion

To some extent it is difficult to discern the individual effect of each change to the baseline model, although we attempt to do so in Table 7.1 where the statistical significance of the individual changes are noted. However, judging from our experiments, we suspect that it is the joint effect of several of the changes taken together that is important in terms of improving accuracy.

Introducing the  $P_{coordWord}$  parameter class for both noun phrases and base noun phrases, and estimating  $P_{coordWord}$  from the word graph, clearly helps disambiguation. In the word graph, data sparseness is decreased because each collected event is made symmetric, and list data from the BNC is included. Storing the billexical data in a graph is a convenient way of conceptualising the data, allows for compact storage, and lends itself easily to measures of word similarity based on vectors. However, the effect of using the similarity function was somewhat disappointing as it increased accuracy only to a small, almost negligible, degree. In §8.2.4 we look at ways the word graph and similarity estimates might be improved upon.

A disadvantage of focusing on coordinate errors within a reranker setting is that,

---

<sup>3</sup>Charniak and Johnson [2005] report an oracle  $f$ -score score of 96.8% whereas [Huang and Chiang, 2005] report an  $f$ -score of less than 94.9% for Collins'  $n$ -best lists.

for validation and testing, results can be obscured somewhat due to the fact that any coordinate structure being evaluated is attached to a whole parse tree and a particular parse tree is chosen as the most probable parse of a sentence due to many factors, not only its coordinate structures. If the only difference between two parses were the coordinate structure this would not be problematic. However this is not always the case. Thus it is possible that some change to the model might result in the correct coordinate structure in a particular parse being assigned a higher weight than other incorrect coordinate structures in the  $n$ -best list of parses, but the reranker nevertheless gives the entire tree, of which the correct coordinate sub-tree is but a component, a lower probability than that assigned some other tree containing an incorrect coordinate structure. To illustrate the point: The oracle set of trees are those trees which score overall highest on precision and recall. In the validation set of top-scoring trees according to our final model there are a total of 130 coordinate noun phrase dependencies which are incorrect and which are correct in the oracle set. Yet there are 17 examples of dependencies in our final set which are correct but which are incorrect in the oracle set. This occurs because in the  $n$ -best list there is no tree which contains both the correct coordinate structure as well as the best scoring syntactic structures for the rest of the tree. This phenomenon means also that an improved coordinate score has somewhat unpredictable effects on the overall parsing  $f$ -score. For example, improvements in coordinate  $f$ -score in the validation set led to no change in the overall  $f$ -score, whereas in section 23 the  $f$ -score rose slightly from 89.4% to 89.6%. This effect could be mitigated if, rather than reranking the top  $n$  trees, the generative model were applied to the full output of the base parser, possible through dynamic programming on a packed representation of the trees. This is a potentially fruitful area of future work.

## 7.5 Summary

Our evaluation of coordinate disambiguation is based on coordinate noun phrase dependencies from section 23 of the Penn WSJ. Although our experiments are focused on making improvements in noun phrase coordination, the context of the experiments is generative reranking - that is each parse tree is given an overall probability based on the full generative parsing model and not just on the score for the coordinate noun

phrases in the tree. We show how various changes in the baseline parsing model, suggested in previous chapters, lead to statistically significant improvements coordination disambiguation in the test set. The overall result on the test set was an increase in coordinate noun phrase *f*-score from 69.91% to 73.77%, which represents a 13% relative reduction in coordinate *f*-score error. In the next and final chapter we conclude and discuss future work that might further improve this score.



# Chapter 8

## Conclusions and Future Work

### 8.1 Summary

In this thesis we have demonstrated how a generative parsing model which uses variations of the  $k$ -nearest neighbour algorithm to estimate the local posterior distributions of the model can achieve state-of-the-art accuracy scores when tested in a reranking setting. By further decomposing the local probability distributions of the base model, enriching the set of conditioning features used to estimate the model's parameters, and using  $k$ -NN as opposed to the Witten-Bell estimation of the base model, we achieve an  $f$ -score of 89.4%, which is a 6% relative decrease in  $f$ -score error over the 1-best output of the base parser. This score was increased to 89.6% when the model was altered to improve its ability to correctly disambiguate coordinate noun phrase structures.

Noun phrase coordination is the worst performing area of the parsing model and a major focus of this thesis was on increasing understanding of coordinate noun phrase ambiguity in order to develop techniques to improve the model's handling of coordinate NP disambiguation. To this effect we gave an empirical analysis of noun phrase coordination in the annotated WSJ. Coordination disambiguation necessitates information from a variety of sources to help make the final disambiguation decision and our multi-faceted approach to improving coordinate noun phrase disambiguation has reflected this fact. We presented a variety of methods which succeeded in increasing coordinate noun phrase dependency  $f$ -score from 69.91% to 73.77% - a 13% relative reduction in coordinate  $f$ -score error.

The remainder of this section outlines in more detail the conclusions of this thesis on memory-based parameter estimation and coordinate noun phrase disambiguation.

### 8.1.1 Memory-Based Parameter Estimation

We have explored a variety of parameter estimation techniques based on  $k$ -NN. We found it useful to employ a constraint feature mechanism whereby the number of examples in the training set used for parameter estimation was restricted according to the values of a set of constraint features. This helped both in terms of accuracy and speed, the latter a particularly important factor in order to make  $k$ -NN a feasible option in the domain of parameter estimation in generative parsing. In addition, we found that combining the  $k$ -NN estimates with the original Witten-Bell estimates improved accuracy for the parameter class used to estimate the probability of constituent head words. We combined the  $k$ -NN and Witten-Bell estimates for this parameter class, backing off to the original Witten-Bell estimate, using a variation of Witten-Bell interpolation where interpolation weights are derived from the count of the constraint features values in the training set.

We also developed a parameter estimation technique for a parameter class which models bilexical coordinate noun data,  $P_{coordWord}(w_i|w_p, H(i))$ . For these estimates there were three levels of back-off. The most specific was a maximum likelihood estimate from data stored in a word graph. The word graph consisted of data not only from the WSJ data set but also from the unannotated BNC data, which helped to decrease data sparsity. In addition, sparseness is decreased because each collected event is made symmetric. Using Witten-Bell interpolation we backed off to a more general estimate which included words similar to  $w_p$  in the training set, again with the aim of reducing data sparsity. This latter estimate was another variation on the  $k$ -NN algorithm, where instead of the inverse of the overlap metric, similarity of two instances is based on a distributional word similarity measure. This estimate was in turn combined with a final layer of backoff where the history was taken to be the POS tag of the word, estimated from WSJ data. The technique used to combine the final two estimates is inspired by Witten-Bell estimation where interpolation weights are derived from the count of each word,  $w_x$ , in the neighbourhood of  $w_p$  and weighted by the similarity of  $w_p$  and  $w_x$ .

Although adding a level of backoff to the estimate which incorporated the similarity function gave only a small increase in accuracy, we believe this is an area of future potential. Though techniques have been developed independently to incorporate measures of lexical similarity into probability estimates (see §2.5), to the best of our knowledge, none have before been integrated into a parsing model. In §8.2.4 we discuss ways the similarity-based estimate might be improved upon.

### 8.1.2 Noun Phrase Coordination Disambiguation

In our analysis of noun phrase coordination in the annotated WSJ we looked at inconsistencies in the annotation of the treebank. We found that noun phrase coordination in the treebank appears to be quite noisy and discussed how this negatively affects coordinate noun phrase disambiguation. We showed how some of these inconsistencies could be detected automatically and how this detection method could be used to clean the data.

We also described a method to facilitate error analysis, which involved aligning incorrect coordinate subtrees, output from the memory-based baseline model, with the correct version of the subtree. We discussed how the differing head-finding rules for noun phrases and base noun phrases are a potential source of coordination error and, in order to minimise this source of error, we made a slight alteration to the head-finding rules for base noun phrases so that the lexicalisation process for coordinate NPBs is more similar to that of other NPs.

Our analysis on parallelism in conjunct structure supports that of previous work [Dubey et al., 2005] and in addition shows that a significant level of parallelism is evident in the syntactic structure of conjuncts at depths greater than one, though decreasing with conjunct depth. We showed how the mechanism used in the empirical analysis for aligning nodes in post-CC conjuncts with nodes in pre-CC conjuncts can be incorporated into the probability model in order to encourage the model to give more weight to syntactic structures which exhibit parallelism.

On semantic similarity between coordinate head nouns we showed that a number of different similarity measures, both distributional and WordNet-based, could detect that nouns that occur together in a coordination pattern are often similar. We also show how the dependencies between conjoined head nouns are not adequately modelled

in the baseline model and developed an alternative that attempts to capture head-head dependencies in both NPs and base NPs.

## 8.2 Future Work

The variations of  $k$ -NN for parameter estimation and Witten-Bell interpolation developed in this thesis were successful at increasing the overall accuracy of parsing model, as well as its coordinate noun phrase accuracy. We believe that parameter estimation might be further improved by further research in a number of areas, outlined in this section. We also present ideas for further improving NP coordination disambiguation and discuss how the generative model can be applied to second-pass parsing and discriminative reranking to improve overall parsing accuracy.

### 8.2.1 Feature Weighting

The parameter estimation techniques presented in this thesis (Chapter 3) did not use feature weighting. Although we did carry out preliminary experiments in information gain and gain ratio feature weighting, they did not improve results. However, intuitively it would seem that some features are clearly more important than others. One possible reason that preliminary experiments in information gain and gain ratio feature weighting did not improve results could be that these are global feature weights. It is possible that more local feature weighting techniques, such as feature weights based on feature values might be more suitable, for example the value difference metric [Stanfill and Walz, 1986] or the modified value difference metric [Cost and Salzberg, 1993], as described in [Daelemans and van den Bosch, 2005, p. 38]. In addition, our feature selection method was a mostly manual hill-climb and an approach to feature selection, such as backward sequential selection, might help ensure that the optimal feature sets for each parameter class are chosen.

### 8.2.2 Constraint Features

It would be interesting to carry out more experiments with the constraint feature mechanism. In particular, we would like to increase the number of constraint feature



sets used. With the current mechanism, depending on the particular feature values, there can be huge differences in the number of samples retrieved at the same level of constraint feature set. For example, the parameter class  $P(C_h|H(i))$  has one constraint feature set,  $\{C_p\}$ , containing one feature  $C_p$  (the nonterminal label of the parent of the node with label  $C_h$ ). If this has the value  $NP$ , 239,018 samples will be retrieved as the initial training set for the  $k$ -NN estimate. If, on the other hand,  $C_p = WHADJP$  only 71 samples will be retrieved. A mechanism whereby, when there are very few samples at the most general level of constraint feature, then another feature is chosen as the constraint feature, might help ensure a minimum number of retrieved samples.

### 8.2.3 Smoothing

For most parameter classes we used the smoothing mechanism for  $k$ -NN of [Toutanova et al., 2003], where in order to avoid zero probabilities we added artificial instances to the training set, one for each class value. It is possible that more sophisticated smoothing might improve the estimates. Combining the  $k$ -NN estimate with the Witten-Bell estimate for the  $P(w_i|H(i))$  parameter class gave an improvement in accuracy over both of the estimates when taken in isolation. It is possible that this approach could work also for the other  $k$ -NN estimates.

### 8.2.4 Word Graph and Similarity Function

There are three main areas of future work which have the potential to improve the word graph-based estimation techniques described in Chapter 4: expanding on and cleaning the data stored in the word graph, improving techniques for the efficient retrieval of the  $k$ -nearest neighbours, and experimenting with different similarity functions.

The data in the word graph could easily be increased, and sparsity thus reduced, by incorporating coordinate events from other unannotated sources as well as more events from the annotated WSJ. Although including data from sources other than the WSJ might be good for parser adaption [McClosky et al., 2006b], results on the WSJ could be improved if unannotated corpora of a similar genre were used. More careful selection of events, similar to the selection of unambiguous coordinations in Goldberg [1999], could also help to reduce noise in the graph. The annotated WSJ is a good (albeit somewhat noisy) source of data. We did not collect head-head data from NPBs

in the WSJ because of possible ambiguities. However, we intend to extract data from unambiguous coordinate NPBs from the annotated WSJ, such as base noun phrases with the following form: (NPB (noun CC noun)). This, as well as including nouns from lists in the hand-parsed WSJ, could help against data sparsity.

In the tests described in Chapter 7, for the estimation of  $P_{sim}(n_i|n_j)$  we searched for the nearest neighbours of  $n_j$  only from among the nodes directly connected to  $n_j$ . In order to expand the search space and possibly return nearer/better neighbours it would be worthwhile to widen the search space as much as is practical, perhaps looking again to a model of retrieval whereby a less costly measure could be used to retrieve an initial set of nearest neighbours to be reduced in turn to  $k$  neighbours by the more sophisticated similarity measure.

In terms of the actual similarity measure used, Lee [1999] compares several different lexical similarity measures for lexical cooccurrence smoothing and found the  $\alpha$ -skew divergence outperformed the cosine similarity metric. Experimenting with different similarity measures would be an interesting area of future work.

Finally, as the word graph estimation technique improved coordination disambiguation, we think it is worth extending to nouns appearing in lists (after all, the data in the graph is part extracted from lists in the BNC). For example, take the phrase:

*(NPB (NNP Paris)(, )(NNP Brussels)(, )(CC and)(NNP Milan))*

Currently, the  $P_{coordWord}(w_i|w_p, H(i))$  parameter class is used to estimate  $P_{coordWord}(w_i = Milan|w_p = Brussels)$ . We would like to investigate using the same parameter class to estimate,  $P_{coordWord}(w_i = Paris|w_p = Brussels)$ .

## 8.2.5 Modelling Dependencies Across CCs

Another information source important to NP coordinate disambiguation is the dependency between non-nominal modifiers and nouns which cross CCs in NPBs. Modelling this type of dependency could help the model learn that the phrase *the cats and dogs* should be bracketed flat, whereas the phrase *the U.S. and Washington* should be given structure. How best one might model these dependencies for parse disambiguation is an open area of research.

### 8.2.6 Cleaning Noisy Data

Results might improve if, rather than merely eliminating noisy events, some of the suspect trees were manually corrected to reflect the Penn guidelines. Correcting the inconsistent coordinate noun phrases which end with the POS pattern *CC/non-nominal modifier/noun*, for example, would involve adding structure, or perhaps merely changing POS tags, for some 202 subtrees.

### 8.2.7 Second-Pass Parsing and Discriminative Reranking

The accuracy of the parsing model described in this thesis might be improved on significantly if, rather than being limited to reranking the top- $n$  trees of a base parser, it were applied to the full output of the base parser. Discriminative rerankers have advantages over history-based approaches in that they are not restricted to choosing features from the parse derivation history but instead can use additional features which incorporate arbitrary aspects of the whole parse tree to improve the initial ranking of the base parser. However, the disadvantage of a generative model in terms of feature selection compared to discriminative rerankers is also its advantage insofar as its limitations on feature selection enable the generative model to use dynamic programming techniques on a packed representation of trees and therefore search over a larger space of possible trees. Applying the generative model as a second-pass of a parser would exploit the full potential of the generative approach.

The generative model outlined in this thesis might also be useful for producing the log probabilities for a discriminative reranker. Our generative model improves the ranking of an initial base parser by recalculating the log probability of each parse produced by the base parser and so produces a more accurate ranking of parses along with their log probabilities. It is possible that improving the log probability ranking of a base parser could improve the scores of the discriminative reranker which uses these log probabilities in its reranking algorithm.

Finally, in future work we will extend our experiments to sentences of all lengths, not only sentences of  $\leq 40$  words. Longer sentences tend to have more conjunctions and therefore the improvements in coordinate noun phrase disambiguation could be especially beneficial for longer sentences.

# Bibliography

- Steven Abney. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4), 1996.
- Rajeev Agarwal and Lois Boggess. A simple but useful approach to conjunct identification. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1992.
- David W. Aha. Generalizing from case studies: A case study. In *Proceeding of the 9th International Conference on Machine Learning (ICML)*, 1992.
- Satanjeev Banerjee and Ted Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceeding of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.
- Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. *Bracketing Guidelines for Treebank II Style Penn Treebank Project*. University of Pennsylvania, 1995. Technical Report.
- Dan M. Bikel. *On The Parameter Space of Generative Lexicalized Statistical Parsing Models*. PhD thesis, University of Pennsylvania, 2004a.
- Daniel M. Bikel. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceeding of the Human Language Technology Conference (HLT)*, 2002.
- Daniel M. Bikel. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4), 2004b.



- Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Proceeding of the 5th Conference on Applied Natural Language Processing*, 1997.
- Ezra Black, Fred Jelinek, John Lafferty, David M. Magerman, Robert Mercer, and Salim Roukos. Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceeding of the 5th DARPA Speech and Language Workshop*, 1992.
- Don Blaheta. Handling noisy training and test data. In *Proceedings of the 7th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.
- Rens Bod. *Beyond Grammar: An Experience-Based Theory of Language*. CLSI Publications, Cambridge University Press, 1998.
- Rens Bod. What is the minimal set of subtrees that achieves maximal parse accuracy? In *Proceeding of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2001.
- Taylor L. Booth and Richard A. Thompson. Applying probability measures to abstract languages. *IEEE Transactions on Computers*, 22(5):442-449, 1973.
- Peter F. Brown, Vincent Della Pietra, Peter deSouza, Jenifer C Lai, and Robert L. Mercer. Class-based  $n$ -gram models of natural language. In *Proceedings of the IBM Natural Language ITL*, 1990.
- Alexander Budanitsky and Graeme Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Proceeding of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLING)*, 2002.
- Lou Burnard, editor. *Users Reference Guide British National Corpus*. Oxford University Computing Services, 1995.
- Sharon Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1999.

- Scott Cederberg and Dominic Widdows. Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of the 7th Conference on Natural Language Learning (CoNLL)*, 2003.
- Eugene Charniak. *Expected-frequency Interpolation*. Brown University, 1996a. Technical Report.
- Eugene Charniak. Tree-bank grammars. In *Proceedings of the 13th National Conference on Artificial Intelligence*, 1996b.
- Eugene Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the 14th National Conference on Artificial Intelligence*, 1997.
- Eugene Charniak. A maximum entropy-inspired parser. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics (NAACL)*, 2000.
- Eugene Charniak and Mark Johnson. Coarse-to-fine  $n$ -best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005.
- Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1996.
- Kenneth Church. Empirical estimates of adaptation: the chance of two Noriegas is closer to  $p/2$  than  $p^2$ . In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, 2000.
- Kenneth W. Church and William A. Gale. A comparison of the enhanced good-turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language*, 5:22-29, 1991.
- Michael Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1996.
- Michael Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and*

*the 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL)*, 1997.

Michael Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

Michael Collins. Discriminative reranking for natural language parsing. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.

Michael Collins and Nigel Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

Michael Collins and Terry Koo. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–69, 2005.

Scott Cost and Steven Salzberg. A weighted nearest neighbour algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.

Walter Daelemans, Antal Van Den Bosch, and Jakub Zavrel. A feature-relevance heuristic for indexing and compressing large case bases. *Poster Papers of the 9th European Conference on Machine Learning*, 1997.

Walter Daelemans, Antal Van Den Bosch, and Jakub Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning, Special Issue on Machine Learning and Natural Language Processing*, 1999a.

Walter Daelemans, Sabine Buchholz, and Jorn Veenstra. Memory-based shallow parsing. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, 1999b.

Walter Daelemans and Antal van den Bosch. *Memory-Based Language Processing*. Cambridge University Press, 2005.

Ido Dagan, Lillian Lee, and Fernando Pereira. Similarity-based models of word co-occurrence probabilities. *Machine Learning*, 34(1-3), 1999.

- Ido Dagan, Fernando Pereira, and Lillian Lee. Similarity-based estimation of word cooccurrence probabilities. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 1994.
- Markus Dickinson and W. Detmar Meurers. Prune diseased branches to get healthy trees! How to find erroneous local trees in a treebank and why it matters. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, Barcelona, Spain, 2005.
- Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7), 1998.
- Amit Dubey, Patrick Sturt, and Frank Keller. Parallelism in coordination as an instance of syntactic priming: Evidence from corpus-based modelling. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNP)*, 2005.
- Jason Eisner. *An Empirical Comparison of Probability Models for Dependency Grammar*. University of Pennsylvania, 1996. Technical Report.
- Ute Essen and Valker Steinbiss. Cooccurrence smoothing for stochastic language modeling. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1992. volume 1.
- Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.
- Dedre Gentner and Kenneth D. Forbus. MAC/FAC: A model of similarity-based retrieval. In *Proceedings of the Cognitive Science Society*, 1991.
- Daniel Gildea. Corpus variation and parser performance. In *Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2001.
- Miriam Goldberg. An unsupervised model for statistically determining coordinate phrase attachment. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1999.



- Joshua Goodman. Global thresholding and multiple-pass parsing. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Parsing (EMNLP)*, pages 11–25, 1997.
- Ralph Grishman and John Sterling. Smoothing of automatically generated selectional constraints. *Human Language Technology*, pages 254–259, 1993.
- James Henderson. Inducing history representations for broad coverage statistical parsing. In *Proceedings of the Joint Meeting of the North American Chapter of the Association for Computational Linguistics and the Human Language Technology Conference*, 2003.
- James Henderson. Discriminative training of a neural network statistical parser. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL)*, 2004.
- Donald Hindle and Mats Rooth. Structural ambiguity and lexical relations. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 229–236, 1991.
- Erhard W. Hinrichs, Julia Bartels, Yasuhiro Kawata, Valia Kordoni, and Heike Telljohann. The Tübingen treebanks for spoken German, English, and Japanese. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin, 2000.
- G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum, editor, *WordNet: An electronic lexical database*. MIT Press, 1998.
- Julia Hockenmaier and Mark Steedman. *CCGbank: User's Manual*. University of Pennsylvania, 2005. Technical Report.
- Deirdre Hogan.  $k$ -NN for local probability estimation in generative parsing models. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*, 2005.

- Deirdre Hogan. Coordinate noun phrase disambiguation in a generative parsing model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, 2007a.
- Deirdre Hogan. Empirical measurements of lexical similarity in noun phrase conjuncts. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, 2007b.
- Liang Huang and David Chiang. Better  $k$ -best parsing. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*, 2005.
- Frederick Jelinek. Self-organized language modeling for speech recognition. *Readings in Speech Recognition*, 1990.
- Frederick Jelinek, John Lafferty, David Magerman, Robert Mercer, Adwait Ratnaparkhi, and Salim Roukos. Decision tree parsing using a hidden derivation model. In *Proceedings of the Human Language Technology Workshop (ARPA)*, 1994.
- Frederick Jelinek and Robert Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, 1980.
- J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics (ROCLING)*, 1997.
- Mark Johnson. PCFG models of linguistic tree representations. *Computational Linguistics*, 24, 1998.
- Slava M. Katz. Estimation of probabilities from sparse data for the language component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401, 1987.
- Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL)*, 2003.

- Terry Koo and Michael Collins. Hidden-variable models for discriminative reranking. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.
- Sandra Kübler. Parsing without grammar - using complete trees instead. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, 2004.
- Sandra Kübler. Toward case-based parsing: Are chunks reliable indicators for syntax trees? In *Proceedings of the 21st International Conference of the on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. Workshop on Linguistic Distances*, 2006.
- Sadao Kurohashi and Makoto Nagao. A syntactic analysis method of long japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4), 1994.
- C. Leacock and M. Chodorow. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An electronic lexical database*. MIT Press, 1998.
- Lillian Lee. *Similarity-Based Approaches to Natural Language Processing*. PhD thesis, Harvard University, 1997.
- Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1999.
- D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML)*, 1998.
- David Magerman. Statistical decision-tree models for parsing. In *Proceedings of the 33th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1995.
- David M. Magerman. *Natural Language Parsing as Statistical Pattern Recognition*. PhD thesis, University of Pennsylvania, 1994.

- David M. Magerman and Mitchell P. Marcus. Pearl: A probabilistic chart parser. In *Proceeding of the European Chapter of the Association for Computational Linguistics (EACL)*, 1991.
- David M. Magerman and Carl Weir. Efficiency, robustness and accuracy in Picky chart parsing. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1992.
- Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 2001.
- David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceeding of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2006a.
- David McClosky, Eugene Charniak, and Mark Johnson. Reranking and self-training for parser adaption. In *Proceedings of the 21st International Conference of the on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 2006b.
- Preslav Nakov and Marti Hearst. Using the web as an implicit training set: Application to structural ambiguity resolution. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNP)*, 2005.
- Siddharth Patwardhan and Ted Pedersen. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL 2006 Workshop: Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together*, 2006.
- Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL)*, 1993.
- Carl Pollard and Ivan Sag. *Head-Driven Phrase Structure Grammar*. Chigaco University Press, Chicago, Illonois, 1994.



- Adwait Ratnaparkhi. A linear observed time statistical parser based on maximum entropy models. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1997.
- Adwait Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, 1998a.
- Adwait Ratnaparkhi. Unsupervised statistical models for prepositional phrase attachment. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING)*, 1998b.
- Adwait Ratnaparkhi, Salim Roukos, and R. Todd Ward. A maximum entropy model for parsing. In *Proceedings of the International Conference on Spoken Language Processing*, 1994.
- Philip Resnik. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.
- Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, pages 95–130, 1999.
- Ellen Riloff and Jessica Shepherd. A corpus-based approach for building semantic lexicon. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1997.
- Brian Roark and Eugene Charniak. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of the 17th International Conference of the on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 1998.
- Mihai Rotaru and Diane J. Litman. Exceptionality and natural language learning. In *Proceedings of the 7th Conference on Natural Language Learning (CoNLL)*, 2003.
- Erik F. Tjong Kim Sang. Memory-based shallow parsing. *Machine Learning Research. Special Issue on Machine Learning Approaches to Shallow Parsing*, 2, 2002.

- Beatrice Santorini. *Part-of-Speech Tagging Guidelines for the Penn Treebank Project*. University of Pennsylvania., 1991. Technical Report.
- Remko Scha, Rens Bod, and Khalil Simaán. A memory-based model of syntactic analysis: Data-Oriented Parsing. *Experimental and Theoretical Artificial Intelligence. Special Issue on Memory-Based Language Processing*, 11(3):409–440, 1999.
- Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.
- Libin Shen, Anoop Sarkar, and Aravind K. Joshi. Using LTAG based features in parse reranking. In *Proceedings of the 8th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2003.
- Craig Stanfill and David Walz. Toward memory-based reasoning. *Communications of the ACM*, 29, 1986.
- Rosmary Stegmann, Heike Telljohann, and Erhard W. Hinrichs. *Stylebook for the German Treebank in VERBMOBIL*, 2000. Technical Report 239.
- Kristina Toutanova, Mark Mitchell, and Christopher D. Manning. Optimizing local probability models for statistical parsing. In *Proceedings of the 14th Conference on Machine Learning (ECML)*, 2003.
- Antal van den Bosch, Ton Weijters, H. Jaap van den Herik, and Walter Daelemans. When small disjuncts abound, try lazy learning: A case study. In *Proceedings of the 7th Belgian-Dutch Conference on Machine Learning*, 1997.
- Julie Weeds. Smoothing using nearest neighbours. In *Proceedings of the 6th UK Special Interest Group for Computational Linguistics (CLUK6)*, 2003a.
- Julie Elizabeth Weeds. *Measures and Applications of Lexical Distributional Similarity*. PhD thesis, University of Sussex, 2003b.
- Gray M. Weiss. A quantitative study fo small disjuncts: Experiments and results. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, 2000.
- Dominic Widdows. *Geometry and Meaning*. CSLI Publications, Stanford, USA, 2004.

- Dominic Widdows and Beate Dorow. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, 2002.
- Ian H. Witten and Timothy C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4), 1991.
- Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 1994.
- Jakub Zavrel and Walter Daelemans. Memory-based learning: Using similarity for smoothing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1997.
- Jakub Zavrel, Walter Daelemans, and Jorn Veenstra. Resolving PP attachment ambiguities with memory-based learning. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, 1997.
- Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *Siam Journal of Computing*, 18:1245-1262, 1989.
- Shaojun Zhao and Dekang Lin. A nearest-neighbor method for resolving PP-attachment ambiguity. In *Proceedings of the 1st International Joint Conference on Natural Language Processing (IJCNLP)*, 2004.