Technical Report

UCAM-CL-TR-530
ISSN 1476-2986

Number 530

UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

Subcategorization acquisition

Anna Korhonen

February 2002

JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted September 2001 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Trinity Hall.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

Series editor: Markus Kuhn

ISSN 1476-2986

Abstract

Manual development of large subcategorised lexicons has proved difficult because predicates change behaviour between sublanguages, domains and over time. Yet access to a comprehensive subcategorization lexicon is vital for successful parsing capable of recovering predicate-argument relations, and probabilistic parsers would greatly benefit from accurate information concerning the relative likelihood of different subcategorisation frames (SCFs) of a given predicate. Acquisition of subcategorization lexicons from textual corpora has recently become increasingly popular. Although this work has met with some success, resulting lexicons indicate a need for greater accuracy. One significant source of error lies in the statistical filtering used for hypothesis selection, i.e. for removing noise from automatically acquired SCFs.

This thesis builds on earlier work in verbal subcategorization acquisition, taking as a starting point the problem with statistical filtering. Our investigation shows that statistical filters tend to work poorly because not only is the underlying distribution zipfian, but there is also very little correlation between conditional distribution of SCFs specific to a verb and unconditional distribution regardless of the verb. More accurate back-off estimates are needed for SCF acquisition than those provided by unconditional distribution.

We explore whether more accurate estimates could be obtained by basing them on linguistic verb classes. Experiments are reported which show that in terms of SCF distributions, individual verbs correlate more closely with syntactically similar verbs and even more closely with semantically similar verbs, than with all verbs in general. On the basis of this result, we suggest classifying verbs according to their semantic classes and obtaining back-off estimates specific to these classes.

We propose a method for obtaining such semantically based back-off estimates, and a novel approach to hypothesis selection which makes use of these estimates. This approach involves automatically identifying the semantic class of a predicate, using subcategorization acquisition machinery to hypothesise conditional SCF distribution for the predicate, smoothing the conditional distribution with the back-off estimates of the respective semantic verb class, and employing a simple method for filtering, which uses a threshold on the estimates from smoothing. Adopting Briscoe and Carroll's (1997) system as a framework, we demonstrate that this semantically-driven approach to hypothesis selection can significantly improve the accuracy of large-scale subcategorization acquisition.

Acknowledgements

I wish to thank my supervisor, Ted Briscoe, for his valuable guidance, dedication and inspiration throughout this project. I am indebted to him for suggesting this avenue of research and for providing insightful and helpful feedback whenever I have requested it. I thank him together with John Carroll for the use of their subcategorization acquisition system.

I am grateful to John Carroll and Diana McCarthy for their practical support and advice. John has made various datasets available and patiently answered my endless queries. Diana has provided valuable feedback with regard to almost everything in this thesis. I have learned a great deal while collaborating with her on hypothesis testing and diathesis alternation detection, and wish to thank her for many stimulating discussions and useful suggestions.

I am grateful to Yuval Krymolowski for interesting discussions he has provided on various aspects of my work, especially on hypothesis testing and evaluation. His helpful comments and constructive criticism have led to many improvements. I am indebted to him - along with Diana - for their repeated assurances that I would finish it in the end. It is thanks to Sabine Buchholz, however, that these assurances were ever realized. Sabine helped at the final stages of this work by performing excellent and thorough proof reading at very short notice.

I would like to thank Bonnie Dorr for kindly making her LDOCE codes available and providing assistance in their use. The work reported in this thesis has also benefited from discussions with Martin Choquette, Genevieve Gorrell, Bill Keller, Olivia Kwong, Maria Lapata, Miles Osborne, Steven Pulman, Hinrich Schütze and Aline Villavicencio. Special thanks go to Mertzi Bergman and Derek Birch for putting joy back into my writing, and to Derek for proof reading the final draft.

The financial support of Trinity Hall, the Cambridge University Computer Laboratory, the Cambridge European Trust and the Finnish Government are gratefully acknowledged.

On a personal note, I wish to thank my many friends in Cambridge and elsewhere who have helped me to keep things in perspective. The support and love of my family are constants that I could not have done without. I thank my parents for their continuous encouragement and Ulla for faithfully keeping track of the midsummer parties I have missed.

My greatest thanks goes, however, to Juha - my husband, friend and cook - who has in these four years turned our Cambridge home into a library. His encouragement, support, help and sense of humour all along can never adequately be put into words.

Contents

List of Acronyms and Abbreviations

ANLT	Alvey Natural Language Tools
AVM	Attribute Value Matrix
BC	Brown Corpus
BHT	Binomial Hypothesis Test
BNC	British National Corpus
CG	Categorial Grammar
CIDE	Cambridge International Dictionary of English
CLE	Core Language Engine
COBUILD	Collins COBUILD English Dictionary
COMLEX	COMLEX Syntax Dictionary
EM	Expectation-Maximisation Algorithm
FN	False Negative
FP	False Positive
FS	Feature Structure
DAG	Directed Acyclic Graph
GATE	General Architecture for Text Engineering
GB	Government and Binding Theory
GPSG	Generalized Phrase Structure Grammar
GR	Grammatical Relation
HMM	Hidden Markov Model
HPSG	Head-Driven Phrase Structure Grammar
KL	Kullback-Leibler Distance
LCS	Lexical Conceptual Structure
LDOCE	Longman Dictionary of Contemporary English
LFG	Lexical Functional Grammar
LKB	Lexical Knowledge Base
LLOCE	Longman Lexicon of Contemporary English
LLR	Log Likelihood Ratio
LOB	Lancaster Oslo-Bergen Corpus
LR	Left-to-Right Parsing
MDL	Minimum Description Length
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimate
MRD	Machine-Readable Dictionary
NLP	Natural Language Processing
NYT	New York Times Corpus
OALD	Oxford Advanced Learner's Dictionary

PCFG	Probabilistic Context-Free Grammar
PCP	Probabilistic Chart Parser
PDT	Prague Dependency Treebank
POS	Part of Speech
PS	Phrase Structure
RC	Spearman Rank Correlation
SCF	Subcategorization Frame
SEC	Spoken English Corpus
SUSANNE	Susanne Corpus
TAG	Tree-Adjoining Grammar
TCM	Tree Cut Model
TP	True Positive
UGC	Unification Categorial Grammar
WN	WordNet
WSD	Word Sense Disambiguation
WSJ	Wall Street Journal Corpus

List of Tables

List of Figures

Chapter 1

Introduction

Research into the automatic acquisition of subcategorization frames (SCFs) from corpora is starting to produce large-scale computational lexicons which include valuable frequency information. However, resulting lexicons indicate a need for greater accuracy. One significant source of error lies in the statistical filtering used for 'hypothesis selection' i.e. for removing noise from automatically acquired SCFs. Although this problem has been widely recognized, it has not been addressed. This thesis builds on earlier work in subcategorization acquisition, taking as a starting point the problem of statistical filtering. Our investigations show that filtering performance is limited by lack of accurate back-off estimates for SCFs. We propose a method of obtaining more accurate, semantically motivated back-off estimates, and a novel approach to hypothesis selection which makes use of these estimates. We demonstrate that this semantically-driven approach can significantly improve large-scale acquisition of SCFs.

This introductory chapter first identifies the need for lexical acquisition (section 1.1). It then introduces the phenomenon of verb subcategorization (section 1.2), establishes its importance for natural language processing (NLP) and linguistic theory (section 1.3), and discusses acquisition of this information automatically from corpus data (section 1.4). Section 1.5 summarises our contribution to the field of subcategorization acquisition. The list of external resources used in our research is given in section 1.6. Section 1.7 includes an overview of the organization of this thesis.

1.1 Automatic Lexical Acquisition

In recent years, the importance of the lexicon has increased in both NLP and linguistic theory. Within NLP, much of the early research focused on isolated 'toy' tasks, treating the lexicon as a peripheral component. These days, the focus is on constructing systems suitable for the treatment of large, naturally-occurring texts. Rich lexical knowledge sources have become crucial for NLP systems dealing with real-world applications. At the same time, the importance of the lexicon has increased for theoretical reasons as within linguistic theory, it has taken on an increasingly central role in the description of both idiosyncratic and regular properties of language.

Obtaining large, explicit lexicons rich enough for computational linguistic use has, however, proved difficult. Manual construction of a large-scale lexicon is a major task involving many years of lexicographic work. The advent of computers has alleviated the work, but the lexicon has correspondingly grown in size. Much of the early work on computational lexicography exploited the information in existing machine-readable dictionaries (MRDs) to solve the acquisition bottleneck. However, as MRDs were written with a human reader in mind, converting these resources into satisfactory computational lexicons proved difficult. Manually built lexicons are prone to errors of omission and commission which are hard or impossible to detect automatically (e.g. Boguraev and Briscoe, 1989). It is also costly to extend these resources to cover neologisms and other information not currently encoded.

Recently, there has developed a growing trend to acquire lexical information automatically from corpus data. This approach avoids the above-mentioned problems, gives access to previously lacking frequency information and enables acquisition of lexical information specific to different sub-languages and domains. Methods for automatic lexical acquisition have been developed for many areas and include syntactic category (Finch and Chater, 1991; Schütze, 1993), collocations (Dunning, 1993; Justeson and Katz, 1995), word senses (Pereira *et al.*, 1993; Schütze, 1992), prepositional phrase attachment ambiguity (Hindle and Rooth, 1993; Lauer 1995), word semantic classes (Zernik, 1989), selectional preferences (Resnik, 1993; Ribas, 1995; Poznanski and Sanfilippo, 1995), diathesis alternations (McCarthy and Korhonen, 1998; Schulte im Walde, 2000; Lapata, 1999, 2000; Stevenson and Merlo, 1999; McCarthy, 2001) and SCFs (e.g. Brent, 1991, 1993; Ushioda *et al.*, 1993; Briscoe and Carroll, 1997; Manning, 1993; Ersan and Charniak, 1996; Carroll and Rooth, 1998; Gahl, 1998; Lapata, 1999; Sarkar and Zeman, 2000). Many of these methods are still under development and need further refinement before they can successfully be applied to large scale lexical acquisition. However, they open up the important possibility of automatically constructing or updating lexicons from textual corpora.

Early methods of lexical acquisition tended to favour purely statistical methods, with the aim of deriving all information from corpus data. Recently there has developed a trend towards use of sources of a priori knowledge that can constrain the process of lexical acquisition (e.g. Gazdar, 1996; Klavans and Resnik, 1996). Although the use of such knowledge may introduce human error it can, if accurate, reduce the overall noise level. A priori knowledge can be probabilistic, when, for example, prior distributions used in lexical acquisition are derived from external sources. It can also be discrete, when it means using predefined categories, such as SCFs, parts-of-speech (POS), or semantic networks to guide the acquisition process. Given that the current conception of a computational lexicon has a firm foundation in linguistic theory, one of the challenges and currently underused approaches in this area is to constrain the acquisition process using linguistic insights (Boguraev and Pustejovsky, 1995).

1.2 Verb Subcategorization

To produce a sentence, it is not enough simply to select the appropriate words and string them together in the order that conveys the meaning relations among them. Not all verbs can appear in all sentences, even when the combinations make sense:

(1) a *Sam put the book on the table*
b **Sam put the book*
c **Sam put on the table*
d **Sam put*

The diverse behaviour of verbs can be explained in terms of subcategorization. Different subcategories of verbs make different demands on their arguments. For example, *put* takes a NP-PP complement (1a), but does not permit NP (1b) or PP (1c) complements, nor an intransitive variant (1d). To be grammatical, *put* requires no fewer than three syntactic arguments: a subject, object and an oblique object.

Subcategorization structures are frequently characterized in terms of syntactic frames called 'subcategorization frames'. These provide generalization over various syntactic contexts required by verbs associated with the same syntactic behaviour. For example, we can use the frame NP-PP to characterize the subcategorization structure in (1a), as well as those in *Sam put the book on the table yesterday* and *John flew the plane to Rome*. More or less specific SCF classifications can be made, depending e.g. on whether the frames are parameterized for lexically-governed particles and prepositions, whether any semantic knowledge is incorporated, and so forth[1].

Fully to define the association between a particular subcategorization structure and a given predicate, however, one must go beyond listing of syntactic frames. Full account of subcategorization requires specifying the number and type of arguments that a particular predicate requires, predicate sense in question, semantic representation of the particular predicate-argument structure, mapping between the syntactic and semantic levels of representation, semantic selectional restrictions or preferences on arguments, control of understood arguments in predicative complements, diathesis alternations, and possibly also further details of predicate-argument structure. We shall introduce in detail this range of phenomena in chapter 2.

1.3 Uses of Subcategorization Information

Multidimensional in nature, verb subcategorization is one of the most complex type of information that a computational lexicon should provide. However, it is arguably also one of the most important type of information. Most recent syntactic theories "project" syntactic structure from the lexicon; thus, access to a comprehensive and accurate subcategorization lexicon is crucial when constraining analysis of natural language. Subcategorization information is essential for the development of robust and accurate parsing technology capable of recovering predicate-argument relations and logical forms. Without it, resolving most phrasal attachment ambiguities or distiguishing arguments from adjuncts is difficult. For parsers using statistical methods to rank analyses, information about relative frequencies of different subcategorizations

[1] Different SCF classifications are discussed and exemplified in chapter 2. In this thesis, we describe SCFs using the labels from Briscoe's classification (2000) (included in Appendix A). Most of these labels (e.g. NP-PP mentioned here) essentially describe the complementation pattern of a verb, assuming that subject is obligatory and, by default, an NP. Where this is not the case, it is explicitly stated.

for a given predicate is also vital. It is required e.g. for lexicalising a probabilistic parser with the aim of improving accuracy of disambiguation (Briscoe and Carroll, 1997; Collins, 1997; Carroll, Minnen and Briscoe, 1998).

Besides parsing, access to accurate subcategorization information can also benefit other domains of NLP, as well as linguistic research. For example, subcategorization (frequency) information can be integrated into dictionaries (e.g. Evans and Kilgarriff, 1995; Gahl, 1998) or annotated corpora (Sarkar and Zeman, 2000) in order to improve their content. It can also be used in psycholinguistic research on sentence processing for approximating lexical preferences (Lapata and Keller, 1998; Lapata *et al.*, 2001). In addition, such information could potentially be used to expand the empirical basis of linguistic theory and increase its predictive power (Levin, 1993).

Knowledge of associations between specific SCFs and predicates can, moreover, aid lexical acquisition. For example, if we identify associations, we can gather information from corpus data about head lemmas which occur in argument slots in SCFs and use the information as input to selectional preference acquisition (Schulte im Walde, 2000; McCarthy, 2001). Selectional preferences are an important part of subcategorization specification, since they can be used to aid anaphora resolution (Ge *et al.*, 1998), speech understanding (Price, 1996), word sense disambiguation (WSD) (Ribas, 1995; Resnik, 1997; Kilgarriff and Rosenzweig, 2000) and automatic identification of diathesis alternations from corpus data (Schulte im Walde, 2000; McCarthy, 2001; Lapata, 1999; Stevenson and Merlo, 1999). Diathesis alternations are in turn important. In recent years they have inspired research in lexicalist grammar theories and lexical representation (e.g. Sanfilippo, 1994; Briscoe and Copestake, 1999), machine translation (Dorr, 1997), natural language generation (Stede, 1998), cross-linguistic studies (Pirrelli *et al.*, 1994), dictionary construction (Dang *et al.*, 1998), verb classification (Dorr, 1997), and lexical acquisition (Ribas, 1995; Poznanski and Sanfilippo, 1995; Korhonen, 1998).

1.4 Subcategorization Acquisition

The first systems capable of automatically learning associations between verbs and a small number of SCFs from corpus data emerged roughly a decade ago (Brent, 1991; 1993). Since then research has taken a big step forward. Subsequent systems targeted a larger set of SCFs and/or collected data on the relative frequencies of different SCF and verb combinations (Ushioda *et al.*, 1993; Manning, 1993; Gahl, 1998; Ersan and Charniak, 1996; Carroll and Rooth, 1998; Lapata, 1999). More ambitious systems have recently been proposed which are capable of detecting comprehensive sets of SCFs and producing large-scale lexicons with appropriate SCF frequency data (Briscoe and Carroll, 1997; Sarkar and Zeman, 2000). The different systems vary greatly according to methods used[2]. Regardless of this, they perform similarly. They mostly gather information about syntactic aspects of subcategorization; do not distinguish between various predicate senses, and have a ceiling on performance at around 80%

[2] We shall in chapter 2 survey the various methods.

token recall[3]. Resulting lexicons thus indicate a need for greater accuracy.

Errors arise in automatic subcategorization acquisition for several reasons. Due to ungrammaticalities of natural language, some noise occurs already in input data. Further errors arise when processing the data, typically in two phases: (i) generating hypotheses for SCFs and (ii) selecting reliable hypotheses for the final lexicon. Analysis of error reveals problems common to different systems. Although it is clear that hypothesis generation requires further improvement, the weakest link of current systems appears to be hypothesis selection.

Hypothesis selection is usually made with a hypothesis test and frequently with a variation of the binomial filter introduced by Brent (1993). The binomial hypothesis test is reported to be particularly unreliable for low frequency associations (Brent, 1993; Manning, 1993; Ersan and Charniak, 1996; Briscoe and Carroll, 1997; Manning and Schütze, 1999). Briscoe and Carroll, for example, note that the performance of their filter for SCFs with less than 10 exemplars is inconclusive. The high number of missing low frequency associations directly affects recall, resulting in poor performance.

This problem with hypothesis selection may overturn benefits gained when increasing the data potential in the hope of detecting a higher number of rare SCFs. Similarly, it may overturn benefits gained from refining hypothesis generation. The problem concerns most subcategorization acquisition systems, since nearly all of them perform hypothesis selection using statistical hypothesis tests. For these reasons, when aiming to improve subcategorization extraction, addressing this problem is critical.

1.5 Our Contribution

The aim of the present thesis is to improve the accuracy of subcategorization acquisition by improving the accuracy of hypothesis selection. All the work reported in this thesis is done using Briscoe and Carroll's (1997) system as a framework for subcategorization acquisition. This system represents the latest phase in the development of SCF acquisition technology. Capable of categorizing over 160 SCF types, it is the most comprehensive system available. We justify our choice further in chapter 2, where we describe this system in detail.

1.5.1 Hypothesis Testing

Although statistical filters have been widely recognized as problematic, the reasons for their poor performance have not been investigated. In this thesis we perform a series of experiments to examine why hypothesis testing in subcategorization acquisition fails to perform as expected. We compare three different approaches to filtering out spurious hypotheses. Two hypothesis tests perform poorly, compared with a simple method which filters SCFs on the basis of their maximum likelihood estimates (MLEs). Our investigation reveals that the reason hypothesis testing does not perform well in this task is that not only is the underlying distribution zipfian, but there also is

[3] Where token recall is the percentage of SCF tokens in a sample of manually analysed text that were correctly acquired by the system. For further explanation, see section 2.5.2.

very little correlation between the conditional distribution of SCFs given the predicate and the unconditional distribution independent of specific predicates. Accordingly, any method for hypothesis selection (whether or not based on a hypothesis test) that involves reference to the unconditional distribution, will perform badly.

1.5.2 Back-off Estimates

Assuming that the unconditional distribution provides accurate back-off estimates[4] for any verbs is roughly equivalent to assuming that all verbs behave similarly in terms of subcategorization. This assumption is challenged by simple observation of verb behaviour. For example, a verb like *believe* occurs mostly with a sentential complement, but the sentential complement frame, in general, is rare. Linguistic research has shown that verbs fall into syntactically and semantically based classes distinctive in terms of subcategorization (e.g. Levin, 1993). More accurate back-off estimates might be obtained by constructing them as specific to such classes.

Semantic verb classes such as Levin's are based, however, on associations between specific SCFs and verb *senses*. Subcategorization acquisition systems are so far capable of associating SCFs with verb *forms* only. We perform experiments with a set of SCF distributions specific to verb form, which show that in terms of SCF distributions, individual verbs correlate more closely with syntactically similar verbs and clearly more closely with semantically similar verbs, than with all verbs in general. The best SCF correlation is observed when verbs are classified semantically according to their predominant sense. On the basis of this result, we suggest classifying verbs semantically according to their predominant sense and obtaining back-off estimates specific to semantic classes. In general terms, we propose using a priori discrete and probabilistic knowledge about (generalizations of) verb semantics to guide subcategorization acquisition.

1.5.3 Semantically Driven Hypothesis Selection

We demonstrate the utility of our proposal by presenting a novel approach to hypothesis selection. We compose semantic verb classes based on Levin classes and implement a technique capable of automatically associating verbs with their respective first sense classes via WordNet (Miller *et al.*, 1993). We choose a few representative verbs from the same semantic class and merge their conditional (verb form specific) SCF distributions to obtain class-specific back-off estimates. Subcategorization acquisition machinery is first used for hypothesis generation and the resulting conditional SCF distribution for a predicate is then smoothed with the back-off estimates of the respective semantic class. A simple method is used for filtering, which sets an empirically defined threshold on the probability estimates from smoothing. This allows examination of the potential of back-off estimates without introducing any problems related to hypothesis tests. We demonstrate that the approach provides an effective way of dealing with low frequency associations and a means of predicting those unseen

[4] By back-off estimates, we refer to SCF "prior" probability estimates used for guiding SCF acquisition is some way.

in corpus data. We demonstrate further that the approach is applicable to large-scale subcategorization acquisition and, when applied to this purpose, it results in significant improvement in performance. Overall, our results show that at the level of hypothesis selection, verb semantic generalizations can successfully be used to guide and structure the acquisition of SCFs from corpus data, which so far has been merely syntax driven.

1.6 External Resources

- **Software** For subcategorization acquisition, we employed Briscoe and Carroll's system with a probabilistic chart parser (Chitrao and Grishman, 1990).
- **Corpora** For subcategorization acquisition experiments, we used 20 million words of the written part of the British National Corpus (BNC) (Leech, 1992). Some gold standards used for evaluation in these experiments were also obtained from 1.2 million word data from the Susanne Corpus (SUSANNE) (Sampson, 1995), Spoken English Corpus (SEC) (Taylor and Knowles, 1988), and Lancaster-Oslo-Bergen Corpus (LOB) (Garside *et al.*, 1987).
- **Lexical Resources** For syntactic verb classes, we employed the Alvey NL Tools dictionary (ANLT) (Boguraev *et al.*, 1987). For semantic verb classes, we employed Levin's verb classification (1993). This resource was used along with the verb hierarchy of WordNet (Miller *et al.*, 1993) version 1.6, Longman Dictionary of Contemporary English (LDOCE) (Procter, 1978), and Dorr's (1997) source of LDOCE codes for Levin classes as aid when associating verbs with semantic classes.

1.7 Overview of Subsequent Chapters

The remaining chapters of this thesis are organized as follows:

Chapter 2 (*Background to Subcategorization Acquisition*) introduces the background and motivation for our work. We discuss the phenomenon and theory of verb subcategorization and the task of constructing a subcategorization lexicon. We review attempts to obtain subcategorization lexicons manually and semi-automatically, and establish why automatic acquisition is needed. We then survey approaches to automatic subcategorization acquisition, discuss the state-of-art performance and the problems which need to be addressed. Finally, we define the scope of our work and introduce the subcategorization acquisition system we employ in our research.

Chapter 3 (*Hypothesis Testing for Subcategorization Acquisition*) examines why hypothesis tests do not perform as expected in subcategorization acquisition. We provide theoretical background on hypothesis testing, review the tests used so far, and discuss the problems reported with them. Experiments are then reported where we compare three different methods of hypothesis selection. Two hypothesis tests perform poorly, compared with a simple method of filtering SCFs on the basis of their MLEs. We discuss reasons for this and note that the lack of accurate back-off estimates for SCFs restricts the performance of hypothesis tests as well as that of other methods of hypothesis selection which rely on these estimates.

Chapter 4 (*Back-off Estimates for Subcategorization Acquisition*) addresses the problem that the unconditional SCF distribution provides poor back-off estimates for SCF acquisition. It investigates whether more accurate estimates could be obtained by basing them on semantic or syntactic verb classes. Experiments are reported which show that in terms of verb form specific SCF distributions, individual verbs correlate more closely with other semantically and syntactically similar verbs than with all verbs in general. The closest correlation is observed between semantically similar verbs. On the basis of this result, we suggest classifying verbs semantically according to their predominant sense and obtaining back-off estimates specific to semantic classes.

Chapter 5 (*A New Approach to Hypothesis Selection*) proposes a method for constructing verb class specific back-off estimates and presents a new semantically motivated approach to hypothesis selection. The latter involves smoothing the conditional SCF distribution for a predicate with back-off estimates of the respective semantic class (i.e. the class corresponding to the predominant sense of the predicate), and using a simple method for filtering which places a threshold on estimates from smoothing. We report experiments which demonstrate that the method can significantly improve the accuracy of SCF acquisition.

Chapter 6 (*Semantically Motivated Subcategorization Acquisition*) refines the novel approach to hypothesis selection outlined in chapter 5 further and applies it to large-scale SCF acquisition. We first relate our work to earlier research on semantically motivated lexical acquisition. We then present the revised approach to hypothesis selection along with a new technique capable of automatically identifying the semantic class of a predicate. The overall approach is evaluated with unknown test verbs. Direct evaluation of the acquired lexicons shows that the semantically-driven approach improves the accuracy of SCF acquisition well beyond that of the baseline approach. Task-based evaluation in the context of parsing shows that the subcategorization probabilities acquired using our approach can improve the performance of a statistical parser. Finally, we discuss possible further work.

Chapter 7 (*Conclusions*) summarises the achievements of our work and suggests directions for future research.

Chapter 2

Background to Subcategorization Acquisition

2.1 Introduction

In this chapter, we discuss the background and motivation for our work. We shall start by describing the linguistic phenomenon of verb subcategorization (section 2.2) and considering its account within linguistic theory (section 2.3). We shall then discuss subcategorization lexicons (section 2.4). We establish the requirements of such resources and survey attempts to obtain them manually and semi-automatically. On the basis of this discussion, we argue that when aiming for an adequate lexicon, automatic acquisition is the avenue to pursue. In section 2.5, we focus on automatic acquisition of subcategorization lexicons. We survey various subcategorization acquisition systems, discuss their performance and highlight the problems which need to be addressed to improve performance. After defining the scope of our work, we end the chapter by introducing the subcategorization system used as a framework in our research.

2.2 The Linguistic Phenomenon of Verb Subcategorization

Subcategorization concerns arguments of a predicate. These may be either obligatory or optional, in which case they should be separated from adjuncts. While arguments are closely associated with the predicate and understood to complete its meaning (2a), adjuncts are understood to complete the meaning of the central predication as a whole (2b).

(2) a *He ate **chocolate***
b *He sat **eating chocolate***

A correct and consistent characterization of the argument-adjunct distinction is crucial both for defining and identifying subcategorization. A variety of criteria have been proposed in linguistic literature to help make the distinction. One well-known criterion is the so-called 'elimination' test (e.g. Somers, 1984), which involves eliminating an element from a sentence and observing whether the remaining sentence is still grammatical. If it is grammatical, the element is classified as an adjunct (or in some cases, an optional argument). Otherwise it is classified as an obligatory argument, as e.g. *in his bag* in (3a).

(3) a *He put the apple* ***in his bag***
b **He put the apple*

Other frequently employed tests involve examining passive, theta roles, selectional restrictions, diathesis alternations, island constraints, linear order of phrases and so forth (see e.g. Matthews, 1981; Somers, 1984; Pollard and Sag, 1987). Many of the standard criteria are, however, subject to exceptions: few cover all cases and some are in conflict with each other. Somers (1984) points out, for example, that the elimination test is complicated by the distinction between syntactic and semantic obligatoriness. A semantically obligatory element may in different circumstances, at the syntactic level, be obligatory (4a), optional (4b) or even necessarily realized by zero (4c):

(4) a *He met somebody* vs. **He met*
b *Don't disturb him, he is reading* (*something*)
c *Our boy can already read* vs. **Our boy can already read something*

In fact, there is a grey area of cases which fall outside traditional classification. Some linguists have addressed this problem by proposing finer-grained distinctions along the argument-adjunct scale (Matthews, 1981; Somers, 1984). Somers (1984), for example, proposes distinguishing between six categories. These include (i) 'obligatory complements' (i.e. arguments), (ii) 'adjuncts' and (iii) 'optional complements', exemplified in (3a), (2b) and (2a) respectively, lexically determined and strongly compulsory (iv) 'integral complements' (e.g. *he doesn't have* ***a chance***), (v) 'middles' (e.g. *he smashed the vase* ***with a hammer***), which lie between obligatory complements and adjuncts, and the extreme type of adjuncts called (vi) 'extraperipherals' (e.g. *he can cook,* ***as you know***), which modify an entire proposition, adjuncts included. Separate criteria are proposed for identification of these six categories. Although approaches such as this explain some previously unclear constructions, they still leave fuzzy boundaries between the different categories.

COMLEX Syntax lexicographers (Meyers *et al.*, 1994) have demonstrated that despite these problems, arguments can be distinguished fairly accurately from adjuncts using five criteria and five heuristics for argument-hood and six criteria and two heuristics for adjunct-hood[1]. These criteria and heuristics are culled mostly from the linguis-

[1] Meyers *et al.* conducted an informal experiment where two human judges made substantially the same argument-adjunct distinctions for a set of 154 phrases using the proposed criteria and heuristics.

tics literature and supplemented with rough generalizations. For example, they state that NPs, PPs headed by *to*, and finite clauses without gaps tend to be arguments, while purpose clauses, PPs and ADVPs expressing place, time and manner are usually adjuncts. They also advise that an argument usually occurs with the verb at significantly higher frequency than with most other verbs, while an adjunct occurs with a large variety of verbs with roughly the same frequency and meaning. Conflicts between the criteria are resolved in various ways. For example, the complement-hood criteria override the adjunct-hood criteria in all but a few well-defined cases, a single complement-hood criterion warrants argument analysis, and so forth.

Given the argument-adjunct distinction, subcategorization concerns the specification, for a predicate, the number and type of arguments which it requires for well-formedness. For example, some verbs take NP complements (e.g. *kill* and *build*), while others do not (*die* and *smile*). Some verbs permit a following *whether*-complement clause (*enquire*, *wonder*), others permit a following *that*-complement clause, while others permit neither (*kill*, *die*) and others permit both (*consider*). Such specification is sensitive to 'grammatical functions' i.e. the specific grammatical roles the arguments can bear when present. For instance, (5) shows (with traditional labels) the grammatical functions involved with the arguments of *give*.

(5) ***Tim***$_{\text{SUBJECT}}$ *gave* ***us***$_{\text{OBJECT}}$ ***a house***$_{\text{SECOND\_OBJECT}}$

Semantically, arguments correspond to participants involved in the event described by the verb. The relationship between a particular participant and an event is characterized by a 'thematic role' (i.e. a 'semantic role'). Thematic roles are traditionally described using a discrete set of labels called, for example, 'theta roles' (e.g. Fillmore 1968, Gruber 1976). The following list includes some of the most frequently used theta roles and the properties usually associated with them:

- **agent** a participant asserted as either doing or causing something, often with volition.
- **patient** a participant being affected.
- **experiencer** a participant asserted as being aware of something.
- **theme** a participant asserted as changing a position or state, or being in a particular position or state.
- **source/goal/location** a participant or location asserted as the starting (source) or ending (goal) point of motion, or place (location) of event.
- **recipient/beneficiary/maleficiary** a participant asserted as receiving (recipient), benefiting from (beneficiary) or being hurt by (maleficiary) something.
- **instrument** a participant asserted as being used for some purpose.

See Meyers *et al.* (1994) for details of this experiment.

According to this classification, *give* has three participants in (5): the agent realized by the subject *Tim*, the recipient realized by the object *us*, and the theme realized by the second object *a house*. The task of associating syntactic arguments of a verb with semantic roles (in the manner just indicated) is called 'linking'.

The ways predicates select their arguments is determined by semantic tendencies they have for these arguments, i.e. 'selectional preferences' (Wilks, 1986) or 'restrictions' (Katz and Fodor, 1964). For example, the two sentences in (6) are syntactically identical, but (6b) is semantically unacceptable as it violates the selectional restriction holding between the verb *wrap* and its object.

(6) a *Mary wrapped* ***the box of chocolates*** *in tissue paper*
b **Mary wrapped* ***the orbit of Neptune*** *in tissue paper*

Although subcategorization usually involves reference to semantic arguments of a predicate, semantic selection is not a necessary requirement. Subcategorization can also concern phrases whose occurrence is obligatory in the local phrasal context of the predicate but are not semantically selected by it. Examples of verbs subcategorising for such phrases are 'subject' and 'object raising' verbs. For instance, the subject of the raising verb *seem* can be either contentful (7a) or pleonastic (7b). Raising verbs contrast with superficially similar 'equi' verbs. While one subcategorized dependent of a raising verb is not assigned a semantic role, all subcategorized dependents of an equi verb are assigned a semantic role. *Seem* is thus a one-place predicate (i.e. subject raising verb), while *try* (7c,d) is a two-place predicate (i.e. subject equi verb). This difference is illustrated in (7e,f). An account of these two verb types falls under the rubric of 'control'.

(7) a ***John*** *seems to drive a Ferrari*
b ***It*** *seems to annoy Tim that John drives a Ferrari*
c *John tries to drive a Ferrari*
d **It tries to annoy Tim that John drives a Ferrari*
e seem′ (drive′ John′ Ferrari′)
f try′ (John′ (drive′ John′ Ferrari′))

The same verb may appear with a variety of arguments related to one another through 'diathesis alternations'. Sentences in (8) exemplify the causative-inchoative alternation, where the same argument slot filler can be associated with different grammatical functions, either with the direct object of the transitive reading (8a) or the subject of the intransitive reading (8b).

(8) a *Robert rolled* ***the ball***
b ***The ball*** *rolled*

Alternations may involve adding, deleting or subtly changing entailments licenced in a particular construction. This can be illustrated with the dative alternation[2]:

(9) a *John gave champagne to Diana* ↔ *John gave Diana champagne*
b *Joe brought a book to Mary* ↔ *Joe brought Mary a book*
c *Bob promised a new bike for Bill* ↔ *Bob promised Bill a new bike*
d **He charged ten pounds for/to Tom* ↔ *He charged Tom ten pounds*
e **Sarah gave a smile to Tony* ↔ *Sarah gave Tony a smile*
f *David brought a Mercedes to the race* ↔ **David brought the race a Mercedes*

(9a) shows the core case of the dative alternation where a volitional agent causes a willing recipient to receive an object. In (9b,c) the meaning is slightly different: the agent intends to give recipient the object which the recipient may or may not receive. In (9c), the intended act of transfer refers to the future. (9d,e) are dative constructions without oblique counterparts. (9e) is, in addition, a metaphorical/idiomatic extension to the construction. (9f) shows a dative construction without the ditransitive variant.

These examples illustrate that similar verbs with slightly different entailments, or the same verb used in different ways or contexts (accompanied by different arguments), can give rise to different alternation variations. Rather than fully productive, alternations appear semi-productive, as exemplified by numerous exceptions to the core constructions, e.g. (9d,e,f).

What we understand as subcategorization in this thesis thus comprises various facts related to the syntax and semantics of predicate-argument structure. Full account of this linguistic phenomenon requires reference to the syntactic and semantic representation of predicate-argument structure, and to the mapping between the two levels of representation. We shall in the present thesis mainly concentrate on syntactic characterization of subcategorization. In this we shall, however, exploit the close link that exists between the syntactic and semantic characterizations.

2.3 Verb Subcategorization in Linguistic Theory

The theoretical account of verb subcategorization has changed dramatically over time due to the trend of "lexicalism", which has affected both semantic and syntactic theory. In what follows, we will provide a general overview to the account of subcategorization first within semantic and then within syntactic theory[3].

[2] These examples are adapted from Briscoe and Copestake (1999) which provides detailed discussion on dative constructions.

[3] Due to the vast amount of research in these areas and the limited scope of our enquiry, we shall be able to provide a very general overview only and shall have to restrict our discussion to certain theoretical frameworks. See the references given in this section for a fuller picture.

$$
\begin{array}{l}
[_{Event}\ \mathrm{GO}_{Loc} \\
\quad ([_{Thing}\ \mathrm{TIM}], \\
\quad\ [_{Path}\ \mathrm{TO}_{Loc} \\
\qquad ([_{Thing}\ \mathrm{TIM}], \\
\qquad\ [_{Position}\ \mathrm{AT}_{Loc}\ ([_{Thing}\ \mathrm{TIM}],\ [_{Thing}\ \mathrm{HOME}])])])]
\end{array}
$$

Figure 2.1: A sample LCS

2.3.1 The Semantic Basis of Subcategorization

Linking

Much semantic research has studied subcategorization from the perspective of linking. Establishing linking between the syntactic and semantic levels of the predicate-argument structure is not always straightforward. The task is especially complicated by diathesis alternations. In the causative-inchoative alternation, for example, the relation between arguments and roles is not transparent. In the causative variant (8a), the subject is an agent and the object is usually a patient. When no explicit cause is present, however, the patient surfaces as subject (8b), despite its apparent lack of agentive behaviour. In contrast, *the soldiers* in (10b) seem perfectly acceptable as agents on their own, but in the causative reading are relegated to object status. Thus no simple solution of assigning agents to subject and patients to object will suffice.

(10) a *The general marched **the soldiers** down the road*

b ***The soldiers** marched down the road*

Examples such as this suggest the need for a fine-grained semantic representation. Essentially, to provide a full account of the semantic basis of predicate-argument structure, a theoretical framework is required which allows for identification of the subtle meaning components involved in verb behaviour, and a sophisticated means of linking these with corresponding syntactic realizations. Recent proposals for such a framework include e.g. those of Jackendoff (1990), Pinker (1989), Dowty (1991) and Levin and Rappaport Hovav (1996).

Jackendoff (1990) and Pinker (1989) adopt a compositional semantics perspective[4]. Jackendoff views semantic representation as a subset of conceptual structure, and proposes decomposition of verbs into 'lexical conceptual structures' (LCSs). LCSs embody 'types', such as **Event**, **Thing** and **State**, and 'primitives', such as CAUSE, GO and BE. Thematic roles tie the argument positions in a LCS to the NPs in the syntactic structure. Linking is thus established between the LCSs and syntactic structures. Semantically similar verbs take similar LCSs, and alternations are determined as mappings between alternating LCSs. Figure 2.1 shows a simple LCS for *Tim went to home*[5].

[4] In compositional semantics, the idea is to construct sentence meaning from the meaning of constituent words and phrases.

[5] This LCS is adapted from (Dorr, 1997).

Pinker proposes decomposing predicates into structures with dominance relationships. Semantic structures embody the primitives GO, BE, ACT and HAVE. Syntactic structures are projected from the underlying semantic structures via linking rules. For example, Pinker provides a structure for transfer predicates like *give* in which the transfer event (GO) is embedded under the giving event (ACT). The dative version of *give*, on the other hand, has an embedded caused ownership event (HAVE). Thus alternations apply to semantic structures in predictable ways, and linking rules govern whether the resulting alternation structures are acceptably realised. Similar behaviour of a group of verbs is explained in terms of a shared semantic component called 'thematic core'.

Dowty (1991) adopts a different approach, not based on predicate decomposition, but on limiting the number of thematic roles to two 'thematic-role-like concepts': proto-agent (p-agt) and proto-patient (p-pat) roles. These are prototypical clusters of entailments that act as semantic defaults. P-agts tend to be volitional, sentient or perceptive, often causing events or movement. P-pats may be incremental themes or stationary, or undergo a change of state or be otherwise causally affected. With individual predicates, particular participants take on p-agt, p-pat or oblique role status based on the number of contributing entailments they share. The argument with the most proto-agent entailments becomes p-agt (and subject), that with the most proto-patient entailments becomes p-pat (and object), and the remaining participants get oblique status. Thus once the proto roles are assigned, linking follows trivially. In this approach, verb meaning is simply expressed as the combination of a predicate-specific relation with the set of valid entailments for each role. Phenomena such as alternations are sensitive to the distinctions in licenced entailments.

Levin and Rappaport Hovav (1996) introduce yet another type of approach, based on further refinement of the nature of the causation factor. According to the Unaccusative Hypothesis, two classes of intransitive verbs exist, the 'unaccusative' and 'unergative', each associated with a different underlying syntactic configuration. This distinction is said to account for various syntactic phenomena. Levin and Rappaport Hovav argue that unaccusativity is syntactically encoded (in terms of internal and external arguments) but semantically determined (for example, in terms of internal and external causation). It results from the application of linking rules sensitive to internal/external causation. Restrictions on various realizations of causative alternations, for example, are attributable to a distinction between internal and external causation. For instance, verbs amenable both to inchoative and causative forms are verbs of external causation that do not require a volitional agent (e.g. *Robert broke the vase* ↔ *The vase broke*). In contrast, non-alternating verbs are verbs of internal causation that do require a volitional agent (e.g. *Robert broke the promise* ↔ **The promise broke*). This approach is not representational: rather, it is compatible e.g. with predicate decomposition[6].

[6] For other proposals on linking see e.g. Grimshaw, 1990; Guerssel, 1986; Hale and Keyser; 1993.

Lexical Semantic Perspective

In recent years, there has been renewed interest and research within semantic theory on the meaning of words themselves, i.e. how lexical semantic properties affect both syntactic behaviour and (compositional) semantic interpretation. We shall discuss here just two examples, Pustejovsky (1991) and Levin (1993)[7].

Pustejovsky discusses examples such as those in (11a,b) where *enjoy* conveys an identical relation of pleasurable experience between the experiencer subject and the event denoted by the verb's object of which the experiencer is agent. In (11a), we need to explain the manner in which the implicit agent of the event-denoting NP *book-writing* is associated with Wodehouse, while in (11b), we need to explain the mechanism which allows *that book* to denote an event of Wodehouse writing or John reading the book. According to Pustejovsky, *enjoy* coerces its artifact-denoting NP object into an event of some type, while the lexical semantic representation of the NP itself determines the broad nature of understood event. For example, the nature of event in (11b) differs from that in *Wodehouse enjoyed the scene.*

(11) a *Wodehouse enjoys book-writing*
b *Wodehouse / John enjoyed that book*

Positing separate lexical entries for the different syntactic realisations of *enjoy* fails to capture the semantic relatedness of these examples. Pustejovsky proposes a theory of lexical semantics called 'the generative lexicon' the better to account for such phenomena. In his generative model compositionality is assumed and lexical entries contain a range of representative aspects of lexical meaning at different levels: 'argument structure', 'event structure', 'qualia structure' and 'lexical inheritance structure'. Event structure, for instance, identifies the event type involved with a verb or phrase, while lexical inheritance structure determines the relation between words in the lexicon. The levels of representation can be connected e.g. via type coercion, and the operation of 'co-composition' is used to perform specialised inference in predefined ways which control the composition of knowledge structures of words in context. The overall model thus captures subtle meaning variations without attempting to enumerate them.

Levin (1993), on the other hand, argues that alternate syntactic realizations are partly predictable on a semantic basis and may have semantic consequences. For instance, verbs participating in the dative alternation exemplified in (9) are typically change of possession verbs. Change of position verbs, however, can only undergo the alternation if they can be interpreted as conveying a change of possession (e.g. *John slid the beer to the table edge* vs. **John slid the table edge a beer*)[8].

Levin points out that although studies of verb semantics have generally acknowledged

[7] See for further related research especially Goldberg (1994). Goldberg has argued, within her theory of the Construction Grammar (Goldberg, 1994), that *constructions* have meanings independent of lexical items. The subcategorization (frame) itself or the construction is said to contribute aspects of the overall meaning. Thus restrictions on realizations of the dative alternation, for instance, arise because of conflicts between the semantics of the dative *construction* and that of particular arguments.

[8] This example is from Briscoe (1991), p. 43. See the reference for further discussion.

the link between the syntax and semantics of verbs, their continued success will depend partly on extensive exploration of verbs' syntactic behaviour. This, she argues, involves looking at verbs' SCFs, their participation in various diathesis alternations, their morphological properties, as well as extended meanings. Drawing on previous research on verb semantics and her own investigation, Levin identifies 79 alternations involving NP and PP complements and classifies over 3200 verbs as members of these alternations. Moreover, she groups the verbs into 191 semantic classes based on their participation in various sets of alternations. Levin's account of verb semantics is thus descriptive, rather than representational (like e.g. Pustejovsky's account). The resulting source is attractive in providing a summary of the variety of theoretical research done and a reference work extensive enough for practical NLP use. We shall describe Levin's work in detail in section 4.2.1 and discuss its relevance for NLP and lexical acquisition later in this thesis (see especially sections 6.2 and 7.2.2).

2.3.2 Subcategorization in Syntactic Theory

Subcategorization and the Development of Lexical Grammar

In early days of syntactic theory, the entire lexicon was treated as a peripheral component, merely as an appendix to a grammar, or a list of basic irregularities (Bloomfield, 1933). Subcategorization was more or less equated with the number and category of arguments related by a predicate. The lexicon would, for example, encode that *donate* in English means 'X causes Y to have Z' and is a ditransitive verb with regular morphology. However, most other facts - such as that the subject of *donate* typically appears before it - were understood as predictable and fairly general statements about English syntax and were stated independently of the lexicon. Over the past decades, however, the lexicon has taken on an increasingly central role in the description of idiosyncratic, subregular and regular properties of language. Consequently, the importance of subcategorization has increased. In recent syntactic theories, subcategorization represents a complex of information critical to the syntactic behaviour of a lexical item.

The development of "lexicalist grammar" was initiated by Chomsky (1970), who proposed that similarities in the structure of deverbal noun phrases and sentences could be expressed in terms of a lexical relationship between the verb and its nominalization. In this new theory of grammar, lexical redundancy rules were used to express the relationship between a verb and a nominal (e.g. *revolve, revolution*). Bresnan (1976, 1982) characterized further lexical regularities within the syntactic framework called Lexical Functional Grammar (LFG). Central grammatical phenomena (such as passivization) were explained within the lexicon. Overall, the role of the lexicon was considerably larger when compared with other approaches at the time, e.g. the Government and Binding Theory (GB) (Chomsky, 1981). The lexical entries were elaborate, with every inflected form given its own lexical entry.

Gazdar *et al.* (1985) continued the line of work with Generalized Phrase Structure Grammar (GPSG). This syntactic framework provided a novel treatment of subcategorization. Simplifying somewhat, subcategorization is specified in GPSG via a feature which indexes lexical items to specific phrase structure (PS) rules, which introduce

$$\begin{bmatrix} \mathrm{A} : \mathbf{boolean}\,\boxed{1} \\ \mathrm{B} : f \\ \mathrm{C} : \boxed{1} \end{bmatrix}$$

Figure 2.2: A sample feature structure

their appropriate syntactic arguments as phrasal sisters. Verbs of different type are listed in the lexicon with appropriate values for the Subcat(egorization) feature. For example, we could have the rule 'VP ↔ V[Subcat 7] NP' which introduces a simple transitive structure (e.g. *Mary ate the apple*) with the Subcat feature 7 on the V node, and every verb in the lexicon which can appear in that structure carries the feature 7 as part of its lexical entry. Operations which affect bounded dependencies (such as passive) are expressed in GPSG in terms of metarules which systematically manipulate VP rules.

Building on the work of GPSG, Pollard and Sag (1987, 1994) proposed a more radically lexicalist syntactic framework called Head-driven Phrase Structure Grammar (HPSG). In this framework, the syntactic component has been drastically reduced. The construction-specific PS rules are abandoned in favour of a small number of more general rules interacting with a richer lexicon to capture syntactic generalizations. This general PS schema builds constituents according to the specifications of Subcat lists projected from lexical entries. Operations which affect bounded dependencies are expressed in terms of lexical operations (rules) which manipulate Subcat values. We shall take a closer look at the treatment of subcategorization within HPSG later in this section.

Further developments of syntactic theory have likewise continued to relocate information in the lexicon: Categorial Grammar (CG) (e.g. Zeevat *et al.*, 1987), Tree-Adjoining Grammar (TAG) (Joshi *et al.*, 1975), and so forth. As the importance of the lexicon has increased within syntactic theory, the role of other components of grammar has declined. In radically lexicalist theories, the syntactic component is reduced to a few general principles concerning the combination of constituents and all the information about categorial indentity and mode of combination of these constituents is projected from individual lexical entries. Thus these theories, instead of building subcategorization requirements in syntax, do exactly the opposite; they locate virtually all syntactic information into the subcategorization requirements of lexical items.

As more information is located in the lexicon, the question of how the lexicon should be represented has become critical. Most lexicalist theories of grammar (e.g. LFG, GPSG, HPSG, CG) use unification- or constraint-based formalisms (e.g. Shieber, 1986) for lexical representation. These formalisms treat syntactic categories as feature structures (FSs). FSs are formally equivalent to directed acyclic graphs (DAGs) and are displayed in attribute-value-matrix (AVM) notation, as shown in figure 2.2. In AVM notation, features are indicated in UPPERCASE type, types in lowercase **boldface** and DAG reentrancy is indicated by coindexing. The information in FSs is combined using unification. Unification of two FS produces a new FS in which the information from both FSs is monotonically combined.

$$\begin{bmatrix} \textbf{verb-cat} \;\; \textbf{complex-cat} \\ \text{RESULT} : \textbf{cat} \\ \text{DIRECTION} : \textbf{direction} \\ \text{ACTIVE} : \boxed{\text{sign}} \end{bmatrix}$$

Figure 2.3: A sample type: a verb category

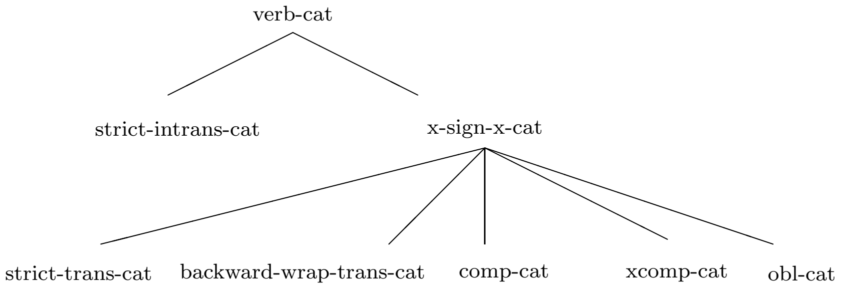

Figure 2.4: A type hierarchy fragment: verb category types

When applying constraint-based formalisms to the lexicon, it is natural to think in terms of typed feature structures (Carpenter, 1992), rather than untyped FSs. The type system may be used to represent the lexicon as an inheritance hierarchy in which information common to a class of lexical items is inherited by all its subclasses. For example, the properties common to all verbs (e.g. POS, presence of a subject) can be defined as a category type which subsumes all members of the verb class. The various subcategories specify different verb types (e.g. intransitive vs. transitive). Figure 2.3 displays a verb category type common to all verbs, and figure 2.4 shows a partial inheritance hierarchy for the sub-types of this type[9]. Although inheritance based on typing is formally attractive, there are linguistic phenomena which involve patterns of regularities and subregularities which cannot be insightfully characterized according to a monotonic inheritance system (Briscoe *et al.*, 1993). Many recent proposals therefore focus on the incorporation of nonmonotonicity i.e. default inheritance (e.g. Carpenter, 1993; Lascarides *et al.*, 1996).

A standard feature of inheritance-based lexicons is the use of lexical rules, i.e. the mappings from one FS to another related one. Lexical rules state conditional implications about the presence of derived lexical entries, given other entries. The rules are used e.g. to represent diathesis alternations. They have taken a variety of forms: see e.g. Shieber (1984), Briscoe and Copestake (1999) and Bresnan and Kanerva (1989).

[9] These examples are taken from Sanfilippo (1993) whose lexical representation is compatible with Unification Categorial Grammar (UCG).

The Grammatical Account of Subcategorization

Grammar theories[10] differ largely in their approach to the argument-adjunct distinction. Various distinctions along the argument-adjunct scale are assumed, and the treatment of the elements classified to the categories adopted varies. Similarly, the number of SCFs assumed and the amount of information provided in them is subject to variation. This is mostly due to diverging dispositions to use syntactic rules and principles to express syntactic generalizations, with a consequent shift of emphasis away from or towards lexical specification. For example, non-lexicalist grammars, such as GB, handle the phenomenon of control in terms of syntactic principles or rules, while lexicalist grammars, such as LFG, HPSG, and CG, encode control in the lexicon, in SCFs of the relevant predicate.

The theories also vary in how they represent the semantics of subcategorization. Some theories employ only one level of syntactic representation and associate a semantic representation with each syntactic constituent in some fashion (e.g. GB, LFG and GPSG). In these theories, argument structure is defined as a level of syntactic description. One such theory is (the early version of) GPSG. It pairs a semantic rule with each syntactic PS rule, which builds the semantics of the left-hand mother category out of the semantics of each right-hand daughter category. Other, more radically lexicalist theories, relocate the semantics directly in the lexicon (e.g. HPSG, CG). In these theories, argument structure is part of the semantic description of the predicates. For example, the lexical entry for a transitive verb includes the information that the semantics of the subject and object syntactic arguments function as the semantic arguments of the predicate associated with the verb. Locating this information (which generalizes to all transitive verbs) in the lexicon allows the semantic representation to build up in tandem with the syntactic representation.

Most syntactic theories approach semantics in compositional manner. The details of the semantic representation, however, vary. Some theories use theta role annotations to rank participants in order to determine their syntactic function. Classifying argument positions into theta roles may be done in terms of traditional classifications of the type introduced in section 2.2. Alternatively, more primitive components of meaning may be assumed, such as those proposed by Pinker (1989) and Dowty (1991) (discussed in section 2.3.1). For example, LFG assumes a hierarchy of traditional theta roles, while UCG makes partial use of Dowty's prototypical roles. GB, instead, uses internal/external argument distinction to determine the structural realization of semantic roles. This distinction is not semantically motivated, but simply assumed as a lexical specification.

Syntactic theories also differ in how they approach linking. Firstly, they vary in how they define and represent grammatical functions. Many currect theories view grammatical functions as links between theta roles and syntactically selected constituents, representing them at the level of lexicon or syntax. Sanfilippo (1990) distinguishes three main orientations according to whether grammatical functions are (i) reduced to constituency relations between phrase markers (as in GB), (ii) defined as primitive elements of the grammar (as in early versions of LFG), or (iii) derived from the

[10] See the previous section for references of grammar theories we discuss in this section.

semantic constituency of predicates (as in CG and HPSG). Secondly, these theories vary in how linking proceeds. For example, in GB, the thematic functionality of argument structure participants is directly projected in syntactic structure. Subjects, objects and other grammatical functions are expressed as predication and government relations between nodes in the tree structures. LFG, instead, uses Lexical Mapping Principles to govern the linking of thematic roles to grammatical function features in lexical forms. In HPSG and CG, arguments are syntactically ranked according to the obliqueness hierarchy which reproposes the grammatical functions in terms of relative position in the Subcat list.

To illustrate the discussion so far, let us consider - as an example - the treatment of subcategorization in (the standard) HPSG. HPSG is a radically lexicalist theory which makes heavy use of unification and where categories incorporate information about the categories they combine with, including subcategorization information. Very few rules are necessary: rather, all important syntactic and semantic processes are driven by information in lexical entries. Much of the PS rules in theories like GPSG are replaced by constraints on the combination (unification) of phrasal and lexical signs. A sign is a FS which encodes PHONology, SYNtax and SEMantic attributes. HPSG makes use of typed signs, organizing the lexicon as an inheritance hierarchy. Distinct verb types (e.g. intransitive and transitive) are characterized by distinct typed signs and subcategorization of verb (sub-)types is encoded in terms of list of categories on the attribute SUBCAT. In lexical entries, the feature SUBCAT is used to encode various dependencies that hold between a lexical head and its complements. The values of this feature contain functional, formal and semantic information, providing uniform treatment of each. Lexical entries can therefore exert restrictions on category selection and government, as well as case and role assignment.

Figure 2.5 shows a simple HPSG style lexical entry for *give*[11]. As illustrated in this entry, the feature SUBCAT takes as its value a list of partially specified SYNSEMs, which bear local values for the attributes CATEGORY and CONTENT. CATEGORY contains information about POS, subcategorization requirements and possible markers, while CONTENT provides information about argument structure. The feature SUBCAT specifies the correspondence between grammatical categories and the semantic roles present at the event described by the verb. The variables associated with the elements of the SUBCAT list unify with the corresponding variables of the semantic roles in the attribute CONTENT. For example, the subject variable (the first element of the SUBCAT list) unifies with the variable filling the 'giver' role.

The flow of subcategorization information up projection paths is handled by the Subcategorization Principle. This principle establishes that the SUBCAT value of a phrase is the SUBCAT value of the lexical head minus those specifications already satisfied by some constituent in the phrase.

HPSG assumes a hierarchy of grammatical categories. Syntactic functions (with the exception of the subject in some versions of HPSG) are defined in terms of the order of corresponding elements on the head's SUBCAT list. The order of this list corresponds to the traditional notion of obliqueness.

[11] The entry is taken from EAGLES (1996), p. 14.

$$\begin{bmatrix} \text{PHON} : \textbf{give} & \\ \text{SYNSEM|LOC|CAT} : & \begin{bmatrix} \text{SUBCAT} : \langle\ \textbf{np}_{\boxed{1}}, \textbf{np}_{\boxed{2}}, \textbf{np}_{\boxed{3}} \rangle & \\ \text{CONTENT} : & \begin{bmatrix} \text{RELN} : \textit{give} \\ \text{GIVER} : \boxed{1} \\ \text{RECEIVER} : \boxed{2} \\ \text{GIVEN} : \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 2.5: HPSG lexical entry for *give*

Operations which involve bounded dependencies (such as passive) or semi-productive diathesis alternations, are expressed in HPSG using lexical operations which manipulate SUBCAT values. They can be captured using lexical rules which map between verb types. For example, to specify passive, a lexical rule may be introduced which removes the first element of a SUBCAT list. Or, to specify an alternation such as the causative-inchoative a lexical rule can be defined which establishes a mapping from the verb type **intrans-verb** to the verb type **trans-causative-verb**, stating a conditional implication about the presence of the 'derived' lexical entry given the basic entry.

In sum, the treatment of subcategorization varies largely from one theoretical framework to another. Within semantic theory there is no consensus regarding the exact meaning components that determine various aspects of verb subcategorization. Rather than being clear, these components appear subtle and elusive. Similarly within syntactic theory, there is no uniform account of this complex phenomenon. However, there is a common trend towards lexicalism, both within semantic and syntactic theory. The importance of the lexicon has increased and at the same time, the importance of subcategorization within the lexicon.

2.4 Subcategorization Lexicon

Given the highly structured conception of the lexicon emerging from linguistic theory, the central role of subcategorization within the theory, the requirements of various theoretical frameworks and the needs of the current (statistical) NLP applications, the question of how to obtain formal, explicit lexicons of sufficiently rich subcategorization has become critical. In what follows, we shall first consider requirements of subcategorization lexicons and then discuss the task of their construction.

2.4.1 Requirements

The design, content and specification of a lexicon for any NLP system is inevitably tied to the purpose for which the NLP system has been constructed, to the influence of prevailing theories and to the current requirements of computational tractability. The lexical knowledge required by different NLP systems ranges from a shallow list of morphological forms to a highly structured and fine-grained lexicon which derives from the linguistic theory adopted. To be practical and useful, however, most NLP systems need a substantial and comprehensive lexicon which covers an adequate vocabulary and encodes the type of qualitative and quantitative knowledge required by the application. A fine-grained lexicon is needed in the increasing number of tasks that require rigorous interpretation of meaning. Some general statements regarding the content of such a lexicon can be found, for example, in Hudson (1995) and Ide and Veronis (1995). In general, the conception of a richer lexicon leads to a combination of morphological, collocational, syntactic, semantic, pragmatic and, for applications involving speech, phonological and phonetic information.

A comprehensive subcategorization lexicon suitable for various NLP uses should firstly distinguish between arguments and adjuncts. This is essential e.g. for a parser, to distinguish between multiple parses of utterances and represent differences in predicate argument structure. Consequences of errors in making this distinction include e.g. generating too few or spurious parses, missing preferences between parses and misinterpreting the predicate argument structure (Meyers *et al.*, 1994).

Given that the argument-adjunct distinction can be established, a subcategorization lexicon must, at the very least, encode the number and category of syntactic arguments associated with different predicates. This information is typically encoded in terms of SCFs. More or less specific SCF classifications have been proposed, depending e.g. on the requirements of a particular syntactic framework assumed. SCFs may e.g. incorporate only syntactic or also semantic information; they may abstract over lexically governed items (such as prepositions and particles) or parameterize for them, and so forth. The fairly detailed classification proposed by Briscoe (2000) (included in Appendix A), for example, incorporates as many as 163 SCF distinctions. It abstracts over specific lexically-governed particles and prepositions and specific predicate selectional preferences, but includes some semi-productive bounded dependency constructions, such as particle and dative movement.

To be compatible with current linguistic theories and guarantee full recovery of logical forms, a subcategorization lexicon should ideally also specify predicate senses, the mapping from syntactic arguments to semantic representation of argument structure, control on predicative arguments, semantic selectional preferences on argument heads, and diathesis alternation possibilities. In addition, it would be important to encode quantitative information, such as the relative frequency of distinct SCFs for each predicate and the probability (or productivity) of various diathesis alternations. This information would be particularly useful to current statistical NLP applications. Knowledge of verb semantic classes or further details of argument structure, such as morphosyntactic properties of arguments, may be useful as well, depending on the intended use of the lexicon.

Both the content and form of a subcategorization lexicon require consideration. As discussed in the previous section, many contemporary grammar theories assume a highly structured organization of a lexicon, which shows convergence of lexical theory and lexicographic practice. This contrasts with the traditional organization of a (subcategorization) lexicon as a list of unrelated lexical entries. The traditional organization lacks generalization and unnecessarily expands lexical representation. In addition, it fails to capture the semantic interrelation between the different verb senses and their corresponding SCFs. According to Levin (1993, p. 1), an ideal lexicon would "provide linguistically motivated lexical entries for verbs which incorporate a representation of verb meaning and which allow the meanings of verbs to be properly associated with the syntactic expressions of their arguments". If a subcategorization lexicon encodes information about alternations and verb semantic classes, this would allow its organization in a compact and linguistically motivated manner (e.g. Sanfilippo, 1994; Briscoe and Copestake, 1999).

Attempts to obtain subcategorization lexicons may be divided into dictionary and corpus-based approaches. We shall discuss these two types of approach in the following sections.

2.4.2 The Dictionary-Based Approach

Several substantial, static subcategorization lexicons exist for English, built either manually by (computational) linguists or largely automatically from machine-readable versions of conventional learners' dictionaries.

Manual construction of lexicons was popular in early stages of NLP. When the systems became more sophisticated and lexicons grew in size, this approach was not entirely abandoned. In the early 1990's, large lexicons or lexical databases (LDBs) were developed, mostly manually, within several projects, e.g. GENELEX, Esprit (Normier and Nossin, 1990); MULTILEX, Esprit (McNaught, 1990). One such substantial subcategorization lexicon is the COMLEX Syntax (Grishman *et al.*, 1994). However, the task of manually developing a large-scale computational lexicon is equivalent to that of developing a conventional advanced learners's dictionary from scratch. It is a major task involving hundreds of years of specification, design, data collection and information structuring - even when assisted by corpus analysis and software support. Not only labour-intensive, manual construction of lexicons leads easily to problems of inconsistency and errors of omission, which are difficult or impossible to detect automatically (Boguraev and Briscoe, 1989).

Since the resources required for manual development of lexicons are typically not available, an alternative approach has, since the 1980's, been to make use of machine-readable dictionaries (MRDs). These include information already categorized, indexed and available in machine readable form. This information may be used to automatically construct a substantial portion of a lexicon, which saves much of the effort involved in manual work. The ideal MRD for this purpose would be a comprehensive advanced learner's dictionary organized as a database. Such a source supplies more grammatical and other information than an ordinary dictionary, as it assumes less linguistic competence on the part of the user (Briscoe, 1991).

Available MRDs, such as LDOCE, COBUILD, the Oxford Advanced Learner's Dictionary (OALD) (Hornby, 1989) or the Cambridge International Dictionary of English (CIDE) (CUP editor, 1995) only, however, approach the ideal. Much effort has been invested in recognizing and compensating for errors and inadequacies in MRDs and/or converting one or several MRDs into a single LDB (e.g. Byrd *et al.*, 1987; Boguraev *et al.*, 1991; Poznanski and Sanfilippo, 1995). This work has been applied to monolingual and bilingual dictionaries, sometimes integrated with language corpora and morphological processing. An example of a substantial subcategorization lexicon constructed from the MRD of LDOCE via some manual intervention is the ANLT dictionary[12].

While work on MRDs has met with some success, it has not resulted in knowledge-rich lexical resources. Based on manual work and originally written with a human reader in mind, the information included in MRDs is often unsystematic. Even after considerable manipulation, customisation and supplementation, these dictionaries contain errors, inconsistencies and circularities difficult to recognise and compensate for.

Briscoe (2001) notes that (semi-)manually developed lexicons tend to show high precision but disappointing recall. When an open-class vocabulary of 35,000 words (Briscoe and Carroll, 1997) was analysed manually for SCF and predicate associations and the result was compared against associations in ANLT and COMLEX, type precision[13] was around 95% for ANLT and COMLEX, while type recall was only around 76% for ANLT and 85% for COMLEX. Thus despite the large volume of lexicographical and linguistic resources deployed, 16-24% of associations between predicates and SCFs were omitted in these lexicons. Briscoe reports that many of the omitted associations are quite unremarkable. For example, when the associations from ANLT and COMLEX were combined, this still left the following sentence types with the verb *seem* unanalyzed:

(12) a *It seemed to Kim insane that Sandy should divorce*
b *That Sandy should divorce seemed insane to Kim*
c *It seemed as though Sandy would divorce*
d *Kim seemed to me (to be) quite clever / a prodigy*
e *(For Kim) to leave seemed to be silly*
f *The issue now seems resolved*

In addition, there are other shortcomings in the content of the lexicons obtained via dictionary-based work. For example, subcategorization lexicons such as ANLT and COMLEX only associate predicate forms (not predicate senses) with SCFs. Although they encode relatively well the syntactic specification of subcategorization, semantic facts and facts at the boundary between syntax and semantics are poorly encoded. Although e.g. information about lexical selection (i.e. the specific lexical requirements that a verb imposes on its subcategorized content, such as details of bound prepositions, particles and complementizers) is included, information about semantic selectional restrictions/preferences is lacking. Similarly, the encoding of diathesis alternations is inadequate. Only information about one or two well-known alternations, such as the dative construction, is included, and information about verb semantic classes is absent. In addition, the mapping from syntactic arguments to semantic argument structure is not fully specified, and quantitative information, e.g. about relative frequency of SCFs given words, is altogether absent.

The organization of current static lexicons does not meet the ideal discussed in the previous section. Although definition of a lexical entry varies from one lexicon to another (e.g. a lexical entry in ANLT associates a particular verb with one SCF only, while COMLEX gathers under one entry all SCFs taken by a particular verb), lexicons

[12] We will introduce ANLT further in section 4.2.2.

[13] See section 2.5.2 for definition of type precision and type recall.

are generally built in the traditional manner, as lists of unrelated lexical entries. A linguistically versatile lexicon design, e.g. that compatible with current grammar theories would require, again, a more thorough encoding of semantic and syntactic-semantic properties of subcategorization than current lexicons employ.

The general problem with both manually developed lexicons and those developed from MRDs is that the information encoded in them is by definition finite. Adding information currently missing in these resources is possible, although costly and time consuming. However, it will not solve the problem inherent in the dictionary-based approach: given that language varies across sub-languages, domains and over time, a fully accurate static lexicon is unattainable in any case. Subcategorization frequencies have been shown to vary across corpus type (written vs. spoken), corpus genre (e.g. financial news text vs. balanced text), and discourse type (single sentences vs. connected discourse) (Carroll and Rooth, 1998; Roland *et al.*, 2000; Roland and Jurafsky, 1998, 2001). Roland and Jurafsky (2001) have showed that much of this variation is caused by the effects of different corpus genres on verb sense and the effect of verb sense on subcategorization. For example, the *attack* and *bill* senses of *charge* have each different set of SCF probabilities. Moreover, the *bill* sense is much more common in e.g. a newswire corpus than a balanced corpus, while the *attack* sense is frequent in a balanced corpus and rare in a newswire corpus. In consequence, *charge* will have different overall SCF frequencies in these two corpora. Thus the relative frequency of a SCF varies depending on the relative frequency of the sense of a word and often SCFs are different under sense extensions. For example, in *she smiled herself an upgrade*, the entire SCF is only available under the extended sense (Briscoe, 2001).

2.4.3 The Corpus-Based Approach

Thus it seems that a once-and-for-all 'universal' lexical resource is a dead weight, and that lexicons should rather be produced on a case-by-case basis. The problems with dictionary-based lexicons have led to attempts to acquire lexical information from corpus data. This approach has become possible during the past decade or so, when sufficiently cheap computation and large enough corpora have become available. Text corpora are a useful source both of qualitative and quantitative lexical information. Frequency information is crucial for many NLP applications and essential to statistical approaches. Along with linguistic information, it is also relevant to the corpus data from which it is acquired. The latter makes it possible to acquire lexical information specific to different sub-languages, eliminating the necessity of viewing the lexicon as static. In the next section we survey attempts to acquire subcategorization information automatically from corpus data.

2.5 Automatic Subcategorization Acquisition

During the past decade, several works have emerged describing methods of automatic subcategorization acquisition (Brent, 1991, 1993; Ushioda *et al.*, 1993; Manning, 1993; Ersan and Charniak, 1996; Briscoe and Carroll, 1997; Carroll and Rooth, 1998; Gahl, 1998; Lapata, 1999; Sarkar and Zeman, 2000). These methods have so far concentrated on the acquisition of very basic subcategorization information: subcategorization frames (SCFs) and their relative frequencies given specific predicates. Although this work has met with some success, more work is needed before large-scale lexicons encoding accurate and comprehensive subcategorization information can be obtained automatically. The research presented in this thesis builds directly on the work already done on subcategorization acquisition. In particular, the problems it addresses stem directly from earlier research. In what follows, we shall accordingly provide a fairly detailed survey of previous research in the topic. We organize our survey as follows: Section 2.5.1 reviews the different methods used for SCF acquisition. Section 2.5.2 looks into evaluation of these methods and describes the performance of existing SCF acquisition systems. It also discusses the problems that need to be addressed when aiming to improve state-of-art performance, and defines the particular problem we address in this thesis. Finally, section 2.5.3 introduces the system we employ in our research as a framework for SCF acquisition.

2.5.1 Methods

All methods of subcategorization acquisition share a common objective: given corpus data, to identify verbal predicates in this data and record the type and/or number of SCFs taken by these predicates. Typically, they proceed in two steps, by (i) generating hypotheses for SCFs and (ii) selecting reliable hypotheses for the final lexicon. Giving a more detailed description of a "typical" learning process is difficult, as the proposed methods vary in different respects. Firstly, they vary in goal. Some systems learn only SCFs, while others also learn relative frequency of SCFs given specific predicates. Secondly, the methods vary as to whether the SCFs are pre-specified or learned, how many SCFs are targeted or learned, and how they are defined. Further, approaches to hypothesis generation vary, depending on whether raw, partially parsed or intermediately parsed corpus data are used as input to the learning process, and how cues for hypotheses are defined and identified. Hypothesis selection is similarly subject to variation. Some systems treat hypothesised SCFs as absolute SCF indicators, while others treat them as probabilistic indicators. The latter systems typically employ a separate filtering component, with filtering frequently performed using statistical hypothesis tests. However, different hypothesis tests and versions of these tests are in use.

We divide the various methods into three groups which we discuss in the subsequent sections. This grouping reflects chronological development from preliminary systems capable of acquiring only a small number of SCFs towards more ambitious systems suitable for large-scale subcategorization acquisition. It also shows how methods have developed with respect to the different factors listed above[14].

[14] This section serves as an overview: the particularly relevant aspects of the SCF acquisition process and those of individual studies will be discussed more thoroughly in the corresponding chapters to follow.

Preliminary Work

Work on automatic subcategorization extraction was initiated by Brent (1991, 1993) who proposed a preliminary method for acquiring just six SCFs from corpus data. The set of SCFs targeted was manually composed and restricted to those involving basic NP, sentential and infinitival phrases. Brent's purpose was only to exploit unambiguous and determinate information in raw (un-tagged) corpora. A number of lexical cues was defined, mostly involving closed class items, which reliably cue verbs and SCFs.

In Brent's system, hypothesis generation proceeds firstly by finding the verbs in the input, and secondly by finding phrases that represent arguments of the verb. Potential verbs are identified by searching the corpus for pairs of words which occur both with and without the suffix *-ing*. A potential verb is assumed a verb unless it follows a determiner or a preposition other than *to*. For example, *was walking* would be taken as a verb, but *a talk* would not. To obtain unambiguous data, verbs occurring in morphological forms other than the stem form and the *-ing* form are ignored. The resulting data are used as input to SCF identification. First, syntactic phrases near a putative verb occurrence are examined and likely verbal arguments indentified using lexical cues. For example, the clause beginning with *that the* is identified as a potential argument of the verb *tell* in *I want to tell him that the idea won't fly* on the basis that pronouns like *him* rarely take relative clauses. Next, putative argument phrases are classified as SCFs. For instance, a phrase is classified as infinitive complement if the string of words immediately right of the verb matches the cue [*to* V] (e.g. I hope *to attend*).

Although Brent uses highly reliable cues, the correspondence between cues and syntactic structure is still not perfect, and the output of the hypothesis generator contains some noise. For example, using Brent's cues, the verb *refer* is wrongly classified as taking an infinitive complement in a sentence such as *I referred to changes made under military occupation.* Brent (1993) addresses the problem by treating the hypotheses as probabilistic rather than absolute indicators of SCFs. He employs a statistical filter for hypothesis selection, which aims to determine when a verb occurs with a particular SCF often enough that all those occurrences are unlikely to be errors. This filter is based on the binomial hypothesis test (BHT) (Kalbfleisch, 1985). It uses the overall error probability that a particular SCF will be hypothesised and the amount of evidence for an association of that SCF with the verb in question to decide which hypotheses are reliable enough to warrant a conclusion[15].

The main problem with Brent's approach is that it generates high accuracy hypotheses at the expense of coverage. Reliant on raw corpus data, the method is dependent on lexical cues. However, for many verbs and SCFs, no such cues exist. For example, some verbs subcategorize for the preposition *in* (e.g. *They assist the police in the investigation*), but the majority of occurrences of *in* after a verb are NP modifiers or non-subcategorized locative phrases (e.g. *He built a house in the woods*). Thus the approach is not extendable to all SCFs and at any rate leads to ignoring a great deal of information potentially available. Use of only unambiguous data means that corpus analysis will be incomplete and no accurate frequency information can be gathered.

[15] A detailed account of this test and its versions is given in chapter 3.

Further Developments

Given the problems of Brent's method, subsequent approaches to SCF acquisition have opted to seek evidence from all examples in corpus data. This has necessitated the use of annotated input data. The approach has been to extract POS tags from corpora and *chunk* (Abney, 1991) the POS tagged data into non-recursive cores of major phrases, e.g. verb groups, bare unpostmodified NPs, PPs and so forth. Chunks extend from the beginning of the constituent to its head, but do not include the post-head dependents, such as complements and trailing adjuncts. For instance, a verbal chunk generally ends with the head lexical verb, so that complements following the verb are excluded. This is illustrated in the following sentence, chunked into NP and VP chunks:

(13) [$_{\text{NP}}$ We] [$_{\text{VP}}$ lack] [$_{\text{NP}}$ the means] [$_{\text{VP}}$ to do] [$_{\text{NP}}$ that]

Essentially, chunking allows factoring data into those pieces of structure which can be recovered without knowledge of the phenomena that we are trying to acquire (i.e. SCFs).

Ushioda *et al.* (1993), Manning (1993), Gahl (1998) and Lapata (1999) represent the first phase of chunking-based SCF acquisition. They all opt for partial parsing via finite state regular expression pattern matching. Parsing is deterministic, and ambiguities in analysis are typically solved using the longest match heuristic: if there are two possible parses that can be produced for the same substring, the parser chooses the longer match. SCF recognition is usually aided by the use of a small number of lexical cues.

Ushioda *et al.* (1993) adopt a POS tagged version of the Wall Street Journal corpus (WSJ) (Marcus *et al.*, 1993) and a finite-state NP parser, which yields information about minimal noun phrases. Their system is capable of recognizing and calculating the relative frequency of six SCFs, the same set as used by Brent. The hypothesis generator first extracts each sentence containing a verb from the tagged corpus. It then chunks the noun phrases using the NP parser and the rest of the words using a set of 16 symbols and phrasal categories (such as VP, PP, sentence initial and final marker, and so forth). A set of nine SCF extraction rules is then applied to the processed sentences. These rules written as regular expressions are obtained through examination of occurrences of verbs in a training text. For instance, the verb *follow* would be assigned a NP complement in the chunked sentence [$_{\text{NP}}$ *John*] [$_{\text{VP}}$ *followed*] [$_{\text{NP}}$ *him*] via a rule which states that NP chunks immediately following the target verb are NP complements, unless themselves immediately followed by a modal, finite verb or base verb.

The output from the hypothesis generator is fairly noisy. The most frequent source of error is in noun boundary detection caused by the simple NP parser (e.g. *give* *[$_{\text{NP}}$ *government officials rights*] *against the press* vs. *give* [$_{\text{NP}}$ *government officials*] [$_{\text{NP}}$ *rights*] *against the press*). The second most frequent source is error in argument-adjunct distinction. Ushioda *et al.* address this problem by using an additional statistical method for hypothesis selection, which enables their system to learn patterns of errors and substantially increase the accuracy of estimated SCFs. It uses regular expressions as filters for detecting specific features of occurrences of verbs and employs multi-dimensional analysis of these features based on log-linear models and Bayes theorem.

Manning (1993) proposes a similar but more ambitious system capable of recognizing 19 distinct SCFs. These SCFs, some of which are parameterized for a preposition, comprise standard frames occurring e.g. in the OALD, LDOCE and COBUILD dictionaries. Corpus data is first tagged using a stochastic POS tagger and a finite state parser is run on the output of the tagger. It parses complements following a verb until a terminator of a subcategorized argument (e.g. a full stop or subordinating conjunction) is reached. The parser includes an NP recogniser and a set of simple rules for SCF identification. It outputs a list of elements occurring after the verb, putative SCFs and statistics on the appearance of the verb in various contexts.

Due to parser mistakes (e.g., the parser invariably records adjuncts as arguments) and skipping (the parser e.g. skips relative clauses and conjunctions whose scope is ambiguous), the resulting hypotheses are noisy. In fact, the hypothesis generator returns nothing or a wrong SCF in the majority of cases. Instead of refining the hypothesis generator further, Manning places more emphasis on hypothesis selection. Hypotheses are evaluated and filtered, following Brent, by BHT. As the hypotheses are more noisy than those generated by Brent's system, Manning refines the BHT by empirically setting higher bounds on the probability of cues being false for certain SCFs. The resulting lexicon encodes information only about SCFs, not their relative frequencies.

Gahl's (1998) and Lapata's (1999) work differs from Ushioda's and Manning's in that they perform SCF acquisition in the context of corpus query systems. Gahl presents an extraction tool for use with the British National Corpus (BNC) (Leech, 1992) which she uses to create subcorpora containing different SCFs for verbs, nouns and adjectives, given the frames expected for each predicate. Gahl's tool is essentially a macroprocessor for use with the Corpus Query Processor (CQP) (Christ, 1994). In the latter, corpus queries are written in the CQP corpus query language, which uses regular expressions over POS tags, lemmas, morphosyntactic tags and sentence boundaries, essentially simulating a chunk parser. Gahl's macroprocessor allows a user to specify which subcorpora are to be created. A user has the choice of 27 searchable SCFs, based on a selection of those occurring in the COMLEX syntax dictionary. One can, for example, search the corpus for the SCF pattern [*verb* NP VP*ing*]. This query returns correct subcategorizations (e.g. *I kept them laughing*) but also gerunds that are not subcategorized.

Gahl identifies several types of error in output, most of which were caused by the partial parser (e.g. unrecognised null or empty categories, ambiguities in PP attachment and so forth). Despite this, she uses no filtering for hypothesis selection. Nor is any experimental evaluation provided which would show how this system performs. Gahl concentrates only on extracting instances of potential SCFs. She mentions that the subcorpora produced by the tool can be used to determine the relative frequencies of SCFs, but reports no work on this.

Lapata (1999) proposes a method similar to Gahl's. She uses the POS tagged and lemmatized version of BNC as an input to Gsearch (Keller *et al.*, 1999), a tool which allows the search for POS tagged corpora for shallow syntactic patterns based on a user-specified grammar and syntactic query. Gsearch combines a parser with a regular expression matcher. In Lapata's approach, a chunk grammar was specified for recognizing the verbal complex and NPs. The aim was to acquire just three SCFs characteristic of the dative and benefactive alternations. The tool was used to extract corpus tokens matching the SCF patterns [*verb* NP NP], [*verb* NP *to* NP] and [*verb* NP *for* NP]. POS tags were retained in the parser's output which was postprocessed to remove adverbials and interjections.

Lapata reports a high level of noise in the output of the hypothesis generator, mostly resulting from the parser, especially, from the use of the longest match heuristic. For example, the parser wrongly identifies instances of the double object frame tokens containing compounds. It also fails with bare relative clauses, NPs in apposition and often with the argument-adjunct distinction. Lapata addresses this problem by post-processing the data. She employs e.g. linguistic heuristics to aid compound noun detection and disambiguation to reduce errors with PP attachment. After postprocessing, the resulting data is still filtered for hypothesis selection. Lapata experiments with a BHT and a filter based on a simple relative frequency cutoff. The latter compares the verb's acquired SCF frequency with its overall frequency in the BNC. Verbs whose SCF relative frequency is lower than an empirically established threshold are disregarded. The SCF (not verb) specific threshold was determined by taking into account for each frame its overall frequency in the COMLEX dictionary.

The approaches surveyed above represent a clear improvement over Brent's approach. Extracting SCF information from chunked data increases the number of cues available and allows also for low reliability cues. Running in linear time, partial parsing is a quick way to seed the SCF acquisition process with some *a priori* grammatical knowledge. The disadvantage, however, is the high level of noise in output, caused by the limitations of partial parsing and the inadequacy of the longest match heuristic. Most approaches discussed above employ filtering for hypothesis selection and rely on its ability to remove noise. This is questionable, however, since the filters applied are not particularly good at handling noise, much of which gets carried over to the system output. Brent e.g. reports poor performance with his BHT filter for low frequency SCFs. Manning and Lapata make the same observation with their BHT filters.

Towards Large-Scale Subcategorization Acquisition

Subsequent work on SCF acquisition has opted for more knowledge-based hypothesis generation. Instead of acquiring SCFs from partially parsed data, recent systems have acquired this information from data parsed using an 'intermediate' parser. Rather than simply chunking the input (as a partial parser does), an intermediate parser finds singly rooted trees:

(14) [$_{\text{S}}$ [$_{\text{NP}}$ He] [$_{\text{VP}}$ [$_{\text{VP}}$ has remained] [$_{\text{AP}}$ very sick]]]

Although such structures are typically built only using POS tag information, they require global coherence from syntax and therefore impose greater grammatical constraint on analysis. An intermediate parser would e.g. detect that the only verb in a sentence must be a VP and does not misanalyse it as part of an NP, as might a partial parser. The intermediate parsers used have been probabilistic. As statistical parsers allow weighting analyses on the basis of training data, they are likely to yield more reliable outcome than the longest match approach used in earlier SCF acquisition work.

Ersan and Charniak (1996) start this era of work by describing a program which gathers statistical information on word-usage and uses these statistics to perform syntactic disambiguation. Learning verbal SCFs, as well as prepositional preferences for nouns and adjectives, is a byproduct of this program. It first collects statistics on individual words in corpus data, then augments a probabilistic context-free grammar (PCFG) with the lexical statistics and finally uses this version of PCFG to parse new data. The resulting data are examined for SCF detection by observing the VP grammar rules which have applied during parsing. The PCFG contains 1,209 rules for expanding verb phrases, which are mapped into the 16 SCFs employed by the system. The SCFs are the same as employed by Manning, but abstract over prepositions. The hypothesis generator proceeds by examining input data and for each verb, recording the VP rule which has applied and the corresponding SCF. For example, if the rule VP $\longrightarrow$ V PRON NP has applied during parsing, this is mapped to the rule VP $\longrightarrow$ V NP NP and further to the ditransitive SCF NP-NP, which is hypothesised for the verb in question. Ersan and Charniak report that the data from the hypothesis generator are fairly noisy due to tagger and parser errors, but provide no qualitative analysis of these errors. To handle the noise, they employ filtering for hypothesis selection. The data, which also encode SCF frequency information, are filtered using BHT. Ersan and Charniak apply this hypothesis test following Manning, with empirically set values for the falsity of certain SCF cues.

Carroll and Rooth (1998) introduce a different approach, a technique based on a robust statistical parser and automatic tuning of the probability parameters of the grammar. They use an iterative approach to estimate the distribution of SCFs given head words, starting from a hand-written headed context-free grammar (CFG) whose core is a grammar of chunks and phrases which includes complementation rules and a large set of n-gram rules. The latter strings phrasal categories together, modeling a finite state machine. A probabilistic version of this grammar is first trained from a POS tagged corpus using the expectation-maximisation (EM) algorithm, an unsupervised machine learning technique (Baum, 1972). Lexicalised event counts (frequency of a head word accompanied by a SCF) are collected, PCFG is lexicalised on rule heads, after which the EM algorithm is run again. The calculation of expectations uses a probabilistic lexicalised weighting of alternative analyses. This allows iteration of the procedure for an improved model. A training scheme is used where the event counts are collected over a segment of corpus, parameters are re-computed and the procedure is repeated on the next segment of corpus. Finally, results from all iterations are pooled to form a single model. This yields the final probability estimates for verb and SCF combinations.

Carroll and Rooth use the SCF classification of the OALD dictionary. Merging it with the SCFs of their grammar, they end up with 15 SCFs. The hypothesis generator outputs information about SCFs and their relative frequencies. Carroll and Rooth report several types of error in the output, most of which are caused by the inability of the chunk/phrase grammar to deal with the argument-adjunct distinction or with constructions where verbs are not directly linked to their complements because of complex conjunctions, ellipses and so forth. These constructs are resolved as intransitives by the robust parser, which leads to their designation as the largest source of error. Despite the noise, Carroll and Rooth do not employ filtering for hypothesis selection, but include all hypotheses generated in the final lexicon (they employ BHT only when obtaining a lexicon for evaluation purposes). An open question is how useful their fairly noisy lexicon would be when used, for example, to aid parsing.

Two large-scale systems targeting a high number of SCFs have been recently proposed by Briscoe and Carroll (1997) and Sarkar and Zeman (2000). Briscoe and Carroll describe a system capable of categorizing 161 different SCFs. This comprehensive set of SCFs was obtained by merging the SCF classifications of the ANLT and COMLEX dictionaries and manually adding into this set new SCFs discovered from the corpus data. While the previous approaches to SCF acquisition employ only syntactic SCFs, Briscoe and Carroll's frames also incorporate semantic information (e.g. about control of predicative arguments).

The system takes as input raw corpus data, which it tags, lemmatises and parses with a robust statistical parser which uses a feature-based unification grammar formalism. This yields intermediate phrase structure analyses. Local syntactic frames are then extracted from the parsed data (including the syntactic categories and head lemmas of constituents) from sentence subanalyses which begin/end at the boundaries of specified predicates. The resulting extracted subcategorization patterns are then classified as SCFs or rejected as unclassifiable on the basis of the feature values of syntactic categories and the head lemmas in each pattern. Although unclassifiable patterns are filtered out, the output from the hypothesis generator is still noisy, mostly due again to parser error. As the parser has no access to lexical information and ranks analyses using a purely structural probabilistic model, there are errors with the argument-adjunct distinction and with certain SCFs, especially those involving semantic distinctions. Briscoe and Carroll employ BHT for hypothesis selection, refining it with *a priori* estimates of the probability of membership in different SCFs. The resulting lexicon incorporates information both on SCFs and their relative frequencies[16].

The SCF extraction method proposed by Sarkar and Zeman (2000) differs from previous work in several respects. It deals with Czech, learns previously unknown (i.e. not predefined) SCFs, and uses a manually derived dependency treebank (Prague Dependency Treebank, PDT; Hajič, 1998) as input data. The system works by reading in the treebank data and considering each tree containing a verb. Within a tree, the set of all dependents of a verb comprises the 'observed frame', while a SCF is the subset of this observed frame. The task of the learning algorithm is to select the subset most likely to be the SCF for a verb, given its observed frame. Essentially, its aim is to identify arguments from among the adjuncts. The hypothesis generator records the frequency of all subsets of each observed frame in treebank data. The subsets are considered from larger to smaller. Large infrequent subsets are suspected to contain

[16] The work we report in this thesis was done using Briscoe and Carroll's system as a framework for SCF acquisition. A more detailed description of this system is given in section 2.5.3.

adjuncts, so they are replaced by more frequent smaller subsets. Small infrequent subsets may have elided some arguments and are rejected. The resulting frequency data serve as input to hypothesis selection.

Sarkar and Zeman use three alternative hypothesis tests: BHT, log likelihood ratio test (Dunning, 1993) and *t*-score (Kalbfleisch, 1985). They apply the tests "recursively". During the first run, only the observed frames are considered. If an observed frame is not selected, one of its subsets is likely to be the SCF. The subset whose length is one member less is selected as successor of the rejected observed frame and it inherits its frequency. Gradually, frequencies accumulate and frames become more likely to survive. The resulting set of frames is classified as SCFs on the basis of POS labels. Sarkar and Zeman report that, with their experiment, the method learned 137 SCFs from corpus data. No further details of these SCFs are given. Sarkar and Zeman do not define their concept of a SCF anyhow, nor specify the distinctions assumed by their classification.

It is clear that manually derived data provide more accurate input to SCF acquisition than automatically parsed data. The use of manually parsed text is, however, not an optimal solution to the knowledge acquisition problem. Treebanks are expensive to build and parsing text manually is arguably more laborious than collecting information on SCFs.

For reasons given earlier, employing intermediate probabilistic parsing in SCF acquisition is an improvement over the use of partial parsing and the longest match heuristic. In sum, we may say that, while the early work minimised noise at the expense of coverage (both in terms of SCFs and data) (Brent, 1991; 1993), the follow-up work maximised coverage at the expense of accuracy (Ushioda *et al.*, 1993; Manning, 1993; Gahl, 1998; Lapata, 1999), and recent work has aimed to maximise both coverage and accuracy. However, at the present state of development, most intermediate parsers still yield fairly noisy output, mainly due to the lack of lexical and semantic information during parsing. As the output from the hypothesis generator is noisy, filtering is needed when aiming for a high accuracy lexicon. Hypothesis selection techniques adopted by recent approaches are similar to those selected in early work. Ersan and Charniak (1996), Briscoe and Carroll (1997), and Sarkar and Zeman (2000) e.g. all employ BHT as originally introduced by Brent (1993) and subsequently followed by Manning (1993) and Lapata (1999). Although different modifications to this test have been proposed, both early and recent approaches report unreliable performance, especially with low frequency SCFs.

In this section, while surveying SCF acquisition systems, we have mentioned errors typical to different systems. In the next section, we turn to quantitative evaluation and consider the overall performance of these systems.

2.5.2 Evaluation and Performance

Methods for Evaluation

SCF acquisition systems are typically evaluated in terms of 'types' or 'tokens' (e.g. Briscoe and Carroll, 1997; McCarthy, 2001). 'Types' are the set of SCFs acquired.

Type-based evaluation involves assessment of the lexical entries in a lexicon. It is usually performed on unseen test data, with a number of randomly selected test verbs. The SCF types acquired are compared with those found in some gold standard. The gold standard is usually obtained either through manual analysis of corpus data, or from lexical entries in a large dictionary. Both approaches have their advantages and disadvantages. Manual construction of a gold standard is time-consuming, but yields an accurate measure when obtained from the data that the system used to acquire the entries. Meanwhile, obtaining a gold standard from a dictionary is quick, but the resulting standard may not be relevant to the test data. This is because dictionaries may contain SCFs absent from the corpus data or miss SCFs present in the corpus data. For example, by merging the lexical entries from the ANLT and COMLEX dictionaries for the verb *add*, we would get nine gold standard SCF types. Not all may be attested in the corpus data: the relatively low frequency SCF PART-NP-PP (*he added in the wine with the herbs*) e.g. could well be missing. On the other hand, the gold standard does not exhaust all the SCF possibilities. For example, the SCF WHAT-S (*he adds what he thinks is right*) is not included, although it is a sound SCF type for *add* and may occur in the corpus data.

'Tokens' are the individual occurrences of SCFs in corpus data. They are evaluated against manually analysed corpus tokens. Evaluation may be performed on the corpus data from which the acquired SCFs were obtained, to estimate the coverage of the training data, i.e. the coverage of the lexicon the system has learned. This indicates e.g. an estimate of the parsing performance that would result from providing a parser with the SCFs acquired. Alternatively, token-based evaluation may be performed on a different corpus to examine how well the acquired information generalizes.

Evaluation is frequently performed using 'precision' and 'recall' (e.g. Briscoe and Carroll, 1997). Obtaining these measures requires recording the number of

- *true positives* (TPs) - correct SCF types or tokens proposed by the system
- *false positives* (FPs) - incorrect SCF types or tokens proposed by the system
- *false negatives* (FNs) - correct SCF types or tokens not proposed by the system

When evaluating SCF information, precision and recall are usually reported over types. 'Type precision' is the percentage of SCFs that the system proposes which are correct (in the gold standard), while 'type recall' is the percentage of SCFs in the gold standard that the system proposes:

$$Type\, precision = \frac{number\ of\ \textsc{tp}s}{number\ of\ \textsc{tp}s + number\ of\ \textsc{fp}s} \tag{2.1}$$

$$Type\ recall = \frac{number\ of\ \textsc{tp}s}{number\ of\ \textsc{tp}s + number\ of\ \textsc{fn}s} \tag{2.2}$$

One can trade off precision and recall to compromise between making a smaller number of sure guesses (high precision) and a bigger number of noisy guesses (high recall). To make mutual comparison of different systems easier, it may be convenient to combine precision and recall in a single measure of overall performance using e.g. the 'F measure':

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.3}$$

With SCF information, recall is sometimes also reported over SCF tokens. 'Token recall' gives the percentage of SCF tokens in entire test data which are assigned correct SCFs by the system.

$$Token\ recall = \frac{number\ of\ \textsc{tp}s}{total\ number\ of\ test\ tokens} \tag{2.4}$$

The systems that record relative frequencies of different verb and SCF combinations often evaluate the accuracy of the resulting probability distributions as well. This is done by comparing the acquired distribution against a gold standard distribution obtained from manual analysis of corpus data. In this, no established evaluation method exists.

The *ranking* of SCFs within distributions has been compared, firstly, by using a simple method proposed by Briscoe and Carroll (1997). This involves calculating the percentage of pairs of classes at positions (n, m) such that $n < m$ in the acquired ranking that are ordered the same in the correct ranking. Briscoe and Carroll call this measure 'ranking accuracy'. Secondly, the ranking has been evaluated using a Spearman rank correlation coefficient (RC) (Spearman, 1904). This involves (i) calculating the ranks for each of the SCF variables separately, using averaged ranks for tied values, and (ii) finding RC by calculating the Pearson correlation coefficient for the ranks. The Pearson correlation coefficient r is calculated from bivariate data $(x_1, y_1), (x_2, y_2)$,..., (x_n, y_n) where the means of the x-values and y-values are $\bar{x}$ and $\bar{y}$ and their standard deviations are s_X and s_Y:

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \frac{x_i - \bar{x}}{s_X} \frac{y_i - \bar{y}}{s_Y} \tag{2.5}$$

RC takes values between -1 and 1, with values near 0 denoting a low degree of association and values near -1 and 1 denoting strong association.

Meanwhile, the *similarity* between acquired and gold standard SCF distributions has been evaluated using cross entropy, a measure familiar from information theory (Cover and Thomas, 1991). The cross entropy of the acquired distribution q with the gold standard distribution p obeys the identity

$$CE(p, q) = H(p) + D(p \| q) \tag{2.6}$$

where H is the usual entropy function and D the relative entropy, or Kullback-Leibler distance (KL). While entropy measures the complexity of the acquired SCF distribution, KL indicates the dissimilarity of the two distributions.

$$D(p\|q) = \sum_{x} p(x) \ln \frac{p(x)}{q(x)} \tag{2.7}$$

KL is always ≤ 0 and reaches 0 only when the two distributions are identical.

The methods discussed so far are used for evaluating SCF acquisition in its own context. However, it is generally agreed that the ultimate demonstration of success is improved performance on an application task. Task-based evaluation may be done, for instance, by examining application performance with and without integrating the SCF information, and seeing how much the integrated information improves performance. With SCF acquisition, task-based evaluation has so far been carried out in the context of parsing and psycholinguistic experiment. We shall describe these experiments in the following section.

Performance

When examining the performance of the SCF acquisition systems we have surveyed, one must remember that they differ in many ways. Variation in the number of target SCFs, test verbs, gold standards, and in the size of test data make direct comparison of different results difficult. However, examining the different results is useful as it reveals the upper limits of performance of the various state-of-art systems.

Table 2.2 shows type precision, type recall and token recall obtained by the current systems, for those systems which report them. F-measure is calculated and shown as well. The second column indicates the number of (target) SCFs; the third shows the number of test verbs employed; the fourth lists the corpus used for learning and testing (table 2.1 provides further information about the different corpora used), and the fifth column gives the size of the test data from which the test verb instances were extracted. The gold standard adopted is listed in the sixth column.

From the approaches listed in table 2.2, those most comparable are Manning (1993), Ersan and Charniak (1996) and Carroll and Rooth (1998). They each target a similar number of SCFs and evaluate the resulting lexicons against entries obtained from the OALD dictionary. When compared by F-measure, Carroll and Rooth outperform the two other approaches, with Ersan and Charniak in turn outperforming Manning's approach. This is not surprising, given that the hypothesis generator employed by Manning is not as sophisticated as those employed by the two other approaches. Manning extracts SCFs from partially parsed data, while the other two approaches opt for intermediate parsing.

The other approaches and results included in this table cannot be compared directly. Brent's (1993) 85 F-measure e.g. was obtained by classifying sentential-complement taking verbs as members of one of the 6 SCFs, while Briscoe and Carroll's (1997) 55 F-measure was obtained by classifying random verbs as members of one of the 161 SCFs. Also, Sarkar and Zeman's (2000) 88% token recall indicates the percentage of SCF tokens assigned a correct argument-adjunct analysis, not a correct SCF type analysis, as with all other approaches. In addition, their result is obtained from manually parsed data (while others use automatically parsed data), which gives them

Corpus	**Size in Words**	**Corpus Type**	**Reference**
Brown Corpus (BC)	1M	balanced	Francis and Kučera, 1989
Wall Street Journal Corpus (WSJ)	1M	newswire	Marcus *et al.*, 1993
New York Times Corpus (NYT)	173M	newswire	Marcus *et al.*, 1993
Susanne Corpus (SUSANNE)	128K	balanced	Sampson, 1995
Spoken English Corpus (SEC)	52K	balanced	Taylor and Knowles, 1988
Lancaster-Oslo-Bergen Corpus (LOB)	1M	balanced	Garside *et al.*, 1987
British National Corpus (BNC)	100M	balanced	Leech, 1992
Prague Dependency Treebank (PDT)	457K	balanced	Hajič, 1998

Table 2.1: Corpora used in SCF acquisition for learning, training and evaluation

Method	**No. of SCFs**	**No. of Verbs**	**Corpus**	**Data Size**	**Gold Standard**	**Type Precision**	**Type Recall**	**F**	**Token Recall**
Brent (1993)	6	63	BC	1.2M	manual analysis	96%	76%	85	-
Ushioda *et al.* (1993)	6	33	WSJ	300K	manual analysis	-	-	-	86%
Manning (1993)	19	40	NYT	4.1M	OALD	90%	43%	58	-
	19	200	NYT	4.1M	manual analysis	-	-	-	82%
Ersan & Charniak (1996)	16	30	WSJ	36M	OALD	87%	58%	70	-
Carroll & Rooth (1998)	15	100	BNC	30M	OALD	79%	75%	77	-
Briscoe & Carroll (1997)	161	7	SUSANNE, SEC, LOB	1.2M	manual analysis	77%	43%	55	81%
	161	14	SUSANNE, SEC, LOB	1.2M	ANLT COMLEX	66%	36%	47	-
Sarkar & Zeman (2000)	137	914	PDT	300K	manual analysis	-	-	-	88%

Table 2.2: Type precision, type recall, F-measure and token recall evaluation of existing SCF acquisition systems

an advantage in evaluation. Examining the different results we may, however, conclude that, regardless of method, there is a ceiling on SCF acquisition performance around 85 F-measure and 88% token recall.

The results achieved when evaluating the accuracy of SCF frequency distributions are even more difficult to compare, as each system is evaluated using a different method. Ushioda *et al.* (1993) do not provide evaluation of SCF frequencies, but simply state that their acquired and gold standard SCF distributions seem very close. Lapata and Keller (1998) evaluate the SCF extraction method described in Lapata (1999). With 20 SCFs and 42 test verbs extracted from 10M words of BNC they report a high correlation of 0.9 with Spearman correlation co-efficient between their acquired SCF ranking and that obtained through manual analysis of corpus data. Briscoe and Carroll (1997) report 81% ranking accuracy with their 7 test verbs (see section 2.5.2 for their evaluation method). Carroll and Rooth (1998) use cross entropy to determine the similarity between SCF distributions acquired by their system and those obtained through manual analysis of corpus data. They perform no large-scale evaluation but report encouraging results with three individual test verbs, whose SCF distributions show an average of 0.36 Kullback-Leibler distance to the gold standard distributions.

Only two approaches perform task-based evaluation. Lapata and Keller (1998) evaluate automatically acquired SCF frequencies obtained using the method described in Lapata (1999) in the context of psycholinguistic experiments on sentence processing. They examine how well the verb biases obtained from completion studies can be approximated by automatically acquired SCF frequencies. The experiments done with 90 test verbs using Garnsey *et al.*'s (1997) metric show that the acquired SCF frequencies classify verbs correctly either as NP-biased or S-biased 58% of the time, as opposed to their 33% baseline and 76% upper bound. A similar but larger experiment reported by Lapata *et al.* (2001) shows comparable results on this binary ranking task.

Briscoe and Carroll (1997) examine whether the SCF frequency information acquired using their system can improve the accuracy of statistical parsing. They report an experiment where they integrate SCF frequency information in a robust statistical non-lexicalised parser. The experiment is performed using a test corpus of 250 sentences from the SUSANNE treebank, and evaluated with the standard GEIG bracket precision, recall and crossing measures (Grishman *et al.*, 1992). While the bracket precision and recall stayed virtually unchanged, the crossing bracket score for the lexicalised parser showed a 7% improvement, which yet turned out not to be statistically significant at the 95% level. However, a different and larger experiment reported by Carroll, Minnen and Briscoe (1998) yields different results. They use a larger test corpus, acquire SCF data from 10 million words of BNC and use a grammatical relation-based (GR) annotation scheme for evaluation (Carroll, Briscoe and Sanfilippo, 1998) which is more sensitive to argument-adjunct and attachment distinctions. The experiment shows that GR recall of the lexicalised parser drops by 0.5% compared with baseline, while precision increases by 9.0%. While the drop in recall proves not to be statistically significant, the increase in precision does. This shows that the SCF frequencies acquired using Briscoe and Carroll's system can significantly improve parse accuracy.

Discussion

While the results achieved with current systems are generally encouraging, the accuracy of the resulting lexicons shows room for improvement. Errors arise in automatic SCF acquisition for several reasons. Due to ungrammaticalities of natural language, some noise already occurs in input data. Further errors arise when processing the data through different phases of hypothesis generation and selection. In section 2.5.1, we mentioned qualitative errors typical to more and less sophisticated SCF acquisition systems. Some of these errors are common to all extant systems, regardless of their sophistication.

With hypothesis generation, the most frequently reported error is the inability of a system properly to distinguish between arguments and adjuncts (e.g. Brent, 1991, 1993; Manning, 1993; Ushioda *et al.*, 1993; Lapata, 1999; Carroll and Rooth, 1998). This makes detection of SCFs involving prepositional phrases especially difficult. Although one can make simple assumptions, for instance, that arguments of specific verbs tend to occur with greater frequency in potential argument positions than adjuncts, problems arise when the judgments of argument-adjunct distinction require a deeper analysis. Many argument-adjunct tests cannot yet be exploited automatically since they rest on semantic judgments that cannot yet be made automatically[17]. One example is the syntactic tests involving diathesis alternation possibilities which require recognition that the same argument occurs in different argument positions. Recognizing identical or similar arguments requires considerable quantities of lexical data or the ability to back-off to lexical semantic classes.

In fact, there is a limit to how far we can get with subcategorization acquisition merely by exploiting syntactic information. As Briscoe and Carroll (1997) point out, the ability to recognize that argument slots of different SCFs for the same predicate share selectional restrictions/preferences would assist recognition that the predicate undergoes specific diathesis alternations. This in turn would assist inferences about control, equi, and raising, enabling finer-grained SCF classifications and yielding a more comprehensive subcategorization dictionary (Boguraev and Briscoe, 1987). In the end, any adequate subcategorization dictionary needs to be supplemented with information on semantic selectional preferences/restrictions and diathesis alternations to provide a full account of subcategorization and to be useful as a lexical resource.

With hypothesis selection, the largest source of errors is the poor performance of the statistical test often employed for filtering out the noise from the system output. The binomial hypothesis test widely used in early as well as recent SCF acquisition work is reported to be particularly unreliable for low frequency SCFs (Brent, 1993; Manning, 1993; Ersan and Charniak, 1996; Briscoe and Carroll, 1997; Manning and Schütze, 1999). Manning, for instance, notes that BHT seems only to select SCFs which are well attested and conversely, does not select SCFs which are rare. Similarly, Ersan and Charniak note that a large number of SCFs only observed once or a few times in data were rejected by their BHT filter. Briscoe and Carroll note that with their system, the majority of errors in SCF acquisition arise because of the statistical filtering process. The performance of their filter for SCFs with less than 10 exemplars is around chance,

[17] Recall our discussion on the argument-adjunct distinction earlier in section 2.2.

and a simple heuristic of accepting all SCFs with more than 10 exemplars would have produced broadly similar results to those generated by use of the filter. The high number of missing low frequency SCFs has a direct impact on recall, resulting in poor performance.

This problem with hypothesis selection may overturn benefits gained when e.g. allowing for large data or low-reliability SCF cues in the hope of detecting a higher number of rare SCFs. Similarly, it may overturn benefits gained from refining hypothesis generation. The problem concerns most SCF acquisition systems, since nearly all perform hypothesis selection using statistical hypothesis tests. For these reasons, the problem of hypothesis selection remains critical to any attempt to improve subcategorization extraction.

In this thesis we report work on improving the hypothesis selection phase of SCF acquisition. All the work reported is done using Briscoe and Carroll's (1997) system as a framework for SCF acquisition. Capable of categorizing over 160 SCF types, which also incorporate semantic information, this system is the most comprehensive SCF extraction system available. By exploiting a robust statistical intermediate parser and a comprehensive SCF classifier, it represents the latest phase in the development of SCF acquisition technology. The evaluation discussed in this section shows that the system performs with accuracy comparable to that of less ambitious extant systems, most of which are limited to a highly restricted set of syntactically based SCFs. Before proceeding further, we shall describe this system in more detail.

2.5.3 Framework for SCF Acquisition

Briscoe and Carroll's (1997) verbal acquisition system consists of six overall components which are applied in sequence to sentences containing a specific predicate in order to retrieve a set of SCFs for that predicate:

1. **A tagger**, a first-order Hidden Markov Model (HMM) POS and punctuation tag disambiguator (Elworthy, 1994). It assigns and ranks tags for each word and punctuation token in sequences of sentences using the CLAWS-2 tagset[18] (Garside *et al.*, 1987).

2. **A lemmatizer**, an enhanced version of the General Architecture for Text Engineering (GATE) project stemmer (Cunningham *et al.*, 1995). It replaces word-tag pairs with lemma-tag pairs, where a lemma is the morphological base or dictionary headword form appropriate for the word, given the POS assignment made by the tagger.

3. **A probabilistic LR parser**, trained on a tree-bank derived semi-automatically from the SUSANNE corpus, returns ranked analyses (Briscoe and Carroll, 1993; Carroll, 1993, 1994) using a grammar written in a feature-based unification grammar formalism which assigns intermediate phrase structure analyses to tag networks returned by the tagger (Briscoe and Carroll, 1995; Carroll and Briscoe, 1996).

[18] CLAWS = The Consistent Likelihood Automatic Word Tagging System.

4. **A pattern extractor** extracts subcategorization patterns, i.e. local syntactic frames, including the syntactic categories and head lemmas of constituents, from sentence subanalyses which begin and end at the boundaries of specified predicates.

5. **A pattern classifier** which assigns patterns to SCFs or rejects them as unclassifiable on the basis of the feature values of syntactic categories and head lemmas in each pattern.

6. **A SCF filter** which evaluates sets of SCFs gathered for a predicate. It constructs putative SCF entries and filters them on the basis of their reliability and likelihood.

At the first stage of the SCF acquisition process, corpus data is tagged using the tagger based on HMM. The HMM model incorporates transition probabilities (the probability that a tag follows the preceding one) and the lexical probabilities (the probability that a word arises for a particular tag). The tagger hypothesises a non-zero probability tag for each word and gives the most probable sequence of tags, given that the sequence of words is determined from the probabilities. It does this using the Forwards-Backward algorithm (e.g. Manning and Schütze, 1999). The CLAWS-2 tagset used by the tagger includes a total of 166 tags for words and punctuation marks. The tagger may return more than one ranked tag per token. The acquisition system filters out all but the highest-ranked tag, trading a small loss in coverage and accuracy for improved runtime space requirements and efficiency, so that large amounts of text can be processed more easily.

At the second stage, the data output by the tagger is lemmatized. During this process, the words are assigned lemmas, their morphological base or dictionary headword forms, based on their POS assignment. In addition to producing a stem or root form for each token, the lemmatizer also produces a normalised affix (e.g. -*ed* for all past participle forms, both regular and known irregulars).

For example, assuming that we build lexical entries for *attribute* and that one of the sentences in our data is (15), the tagger returns (16) and the lemmatizer returns (17).

(15) `He attributed his failure, he said, to no-one buying his books.`

(16)
```
he_PPHS1 attributed_VVD his_APP$ failure_NN1 ,_, he_PPHS1 said_VVD ,_,
to_II no-one_PN buying_VVG his_APP$ books_NN2
```

(17)
```
he_PPHS1 attribute_VVD his_APP$ failure_NN1 ,_, he_PPHS1 say_VVD ,_,
to_II no-one_PN buy_VVG his_APP$ book_NN2
```

At the third stage of the SCF acquisition process, the tagged and lemmatized data are parsed. The probabilistic parser employed by the system uses a grammar which consists of a 455 phrase structure rule schemata. This grammar is a syntactic variant of a Definite Clause Grammar (DCG; Pereira and Warren, 1980) with iterative

Kleene operators. It is shallow, which means that no attempt is made fully to analyse unbounded dependencies. However, the distinction between arguments and adjuncts is expressed, following X-bar theory (e.g. Jackendoff, 1977) by Chomsky-adjunction to maximal projections of adjuncts (XP → XP Adjunct) as opposed to government of arguments (i.e. arguments are sisters within X1 projections; X1 → X0 Arg1... ArgN). All analyses are rooted in S so the grammar assigns global, intermediate and often 'spurious' analyses to many sentences. There are 29 different values for VSUBCAT and 10 for PSUBCAT[19] , which are later analysed along with specific closed-class head lemmas of arguments (e.g. *it* for dummy subjects) to classify patterns as evidence for one of the SCFs. Currently, the coverage of this grammar, the proportion of sentences for which at least one analysis is found, is 79% when applied to the SUSANNE corpus. Wide coverage is important here because information is acquired only from successful parses.

The parser ranks analyses using a purely structural probabilistic model, which makes training the parser on realistic amounts of data and using it in a domain-independent fashion feasible. The model is a refinement of PCFG conditioning context free backbone application on left-to-right (LR) state and lookahead item. Probabilities are assigned to transitions in the LR action table via a process of supervised training. The latter is based on computing the frequency with which transitions are traversed in a corpus of parse histories. The parser is capable of probabilistically discriminating derivations which differ only in terms of order of application of the same set of CF backbone rules, due to the parse context defined by the LR table.

(18) illustrates the highest ranked analysis the parser would return for the lemmatized sentence exemplified in (17).

(18)
```
(Tp
  (V2 (N2 he_PPHS1)
  (V1 (V0 attribute_VVD))
      (N2 (DT his_APP$)
          (N1
            (N0 (N0 failure_NN1)
                (Ta (Pu ,_,)
                    (V2 (N2 he_PPHS1)
                    (V1 (V0 say_VVD))) (Pu ,_,)))))
      (P2
          (P1 (P0 to_II)
            (N2 no-one_PN)
            (V1 (V0 buy_VVG)
                (N2 (DT his_APP$) (N1 (N0 book_NN2)))))))))
```

Quite often the parser has no mechanism for choosing the correct analysis and hence the output is noisy. This is illustrated in example (19), where the correct analysis for (19a) is shown in (19c) and the correct analysis for (19b) in (19d) (Briscoe, 2001).

[19] VSUBCAT stands for 'verbal' subcategorization and PSUBCAT for 'prepositional'.

(19) a `He looked up the word`

b `He looked up the hill`

c `(Tp (V2 (N2 he_PPHS1) (V1 (V0 (V0 look_VVD) (P0 up_RP)) (N2 (DT the_AT) (N1 (N0 word_NN1)))))`

d `(Tp (V2 (N2 he_PPHS1) (V1 (V0 look_VVD) (P2 (P1 (P0 up_RP) (N2 (DT the_AT) (N1 (N0 hill_NN1))))))`

The parser cannot reliably select between (19c) and (19d) because it has no access to any lexical information. In this case it has no information about the likelihood of *look up* being a phrasal verb nor the differing selectional restrictions on the NP as either PP or verbal argument.

At the fourth processing stage, the extractor takes as input analyses from the parser. It extracts subcategorization patterns by locating the subanalyses around the predicate and finding the constituents identified as complements inside each subanalysis and the subject preceding it. Passive constructions are treated specifically. The extractor returns the predicate, the VSUBCAT value and the heads of the complements. In case of PPs, it returns the PSUBCAT value, the preposition head and the heads of the PP's complements.

For example, taking as input the analysis shown in (18), the extractor would yield the subcategorization pattern exemplified in (20).

(20)
```
((((he:1 PPHS1)) (VSUBCAT NP_PP) ((attribute:6 VVD) ((failure:8 NN1))
((PSUBCAT SING) ((to:9 II)) ((no-one:10 PN)) ((buy:11 VVG)))))
```

At the fifth stage, the extracted subcategorization patterns are fed into the pattern classifier, which assigns the patterns into SCFs. The SCFs used in the system were constructed by manually merging the SCFs of the ANLT and COMLEX syntax dictionaries and adding around 30 SCFs found by examining unclassifiable patterns of corpus examples. These consisted of some extra patterns for phrasal verbs with complex complementation and flexible ordering of the preposition or particle, some for non-passivizable patterns with a surface direct object, and some for rarer combinations of governed preposition and complementizer combinations. The resulting set of SCFs abstract over specific lexically-governed particles and prepositions and specific predicate selectional preferences. However, they include some derived semi-predictable bounded dependency constructions, such as particle and dative movement. The current version of the classification comprises 163 SCFs (Briscoe, 2000) and is included in Appendix A of this thesis.

The classifier provides translation between extracted SCF patterns and the two existing dictionaries and a definition of the target subcategorization dictionary. It assigns subcategorization patterns into classes on the basis of the VSUBCAT and PSUBCAT values and sometimes also the lexical information included in patterns. For example, the subcategorization pattern exemplified in (20) is classifiable as the SCF NP-P-NP-ING (transitive plus PP with non-finite clausal complement) with additional lexical information, such as the preposition and the heads of the NP arguments and of the NP and VP arguments of the PP. Each SCF is represented as a SCF class number. In this case the classifier returns two SCFs, 43 and 44. (21) shows the entries for these SCFs in the classification. The first line of an entry shows the COMLEX SCF name, the second gives the frame specification according to ANLT, the third shows a tagged example sentence where the frame occurs, and the final line gives the SCF specification according to the grammar employed by the system[20].

(21)
```
43. NP-P-NP-ING / ??
   ANLT gap (SUBCAT NP_PP_SING)
he_PPHS1 attributed_VVD his_AT failure_NN1 to_II no-one_NP1 buying_VVG
his_AT books_NN2
   (VSUBCAT NP_PP) to (PSUBCAT SING)

44. NP-P-POSSING / ??
   ANLT gap (SUBCAT NP_PP_SING)
They_PPHS2 asked_VVD him_PPHO1 about_II his_PPHO1 participating_VVG
in_II the_AT conference_NN1
   (VSUBCAT NP_PP) about (PSUBCAT SING)
```

More than one SCF is returned by the classifier when it cannot tell which of the SCFs is the correct one. In this case, SCF 43 provides the correct analysis, but the classifier cannot distinguish it from the similar SCF 44, due to the parser problems discussed above.

The classifier also filters out as unclassifiable around 15% of patterns. These are spurious analyses output by the extractor which do not conform to the known SCFs for English. Additionally, as the parser output is noisy, many classifiable patterns are still incorrect and hypothesis selection is needed.

At the final processing stage, the system employs a filter for hypothesis selection. The filter first builds putative lexical entries specific to the verb and SCF combinations. It takes the patterns for a given predicate built from successful parses and records the number of observations with each SCF. Patterns provide several types of information which can be used to rank or select between them, such as the ranking of the parse from which it was extracted or the proportion of subanalyses supporting a specific pattern. Currently, the system simply selects the pattern supported by the highest ranked parse. The resulting putative SCF entries for a predicate are filtered using the binomial hypothesis test.

BHT attempts to determine whether one can be confident that there is a genuine association between a hypothesised verb and SCF combination. The test uses the overall error probability that a particular SCF (scf_i) will be hypothesised, and the amount of evidence for an association of scf_i with the predicate form in question. The error probability for a given scf_i is estimated by

$$p^e = \left(1 - \frac{|verbs\ in\ scf_i|}{|verbs|}\right) \frac{|patterns\ for\ scf_i|}{|patterns|} \tag{2.8}$$

[20] See Appendix A for full details of these entries.

where the counts for SCFs were obtained by running the system's pattern extractor on the entire SUSANNE corpus and the counts for verbs associated with SCFs were obtained from the ANLT dictionary. The probability of an event with probability p happening exactly m out of n attempts is given by the binomial distribution:

$$P(m, n, p) = \frac{n!}{m!(n-m)!} p^m (1-p)^{n-m} \tag{2.9}$$

The probability of the event happening m or more times is:

$$P(m+, n, p) = \sum_{k=m}^{n} P(k, n, p) \tag{2.10}$$

So $P(m+, n, p^e)$ is the probability that m or more occurrences scf_i will be associated with a predicate occurring n times. A threshold on this probability is set at 0.05, yielding a 95% confidence that a high enough proportion of patterns for scf_i have been seen for the verb to be assigned scf_i.

The resulting lexicon is organized by verb form with sub-entries for each SCF. (22) shows a putative lexical entry built for *attribute*, given the subcategorization pattern shown earlier in (20) and the SCF assignment in (21). The entry, displayed as output by the system, includes several types of information. In addition to specifying the verb and SCF combination in question and its frequency in corpus data, it specifies the syntax of detected arguments, the reliability of the entry according to the parser and the value assigned to it by BHT. It also gathers information about the POS tags of the predicate tokens, the argument heads in different argument positions and the frequency of possible lexical rules applied. The different fields of the entry are explained in the legend below it. For example, the entry in (22) indicates that *attribute* was observed in the data only once with the SCF 43 44 (:FREQCNT 1), and therefore the entry gathers information from only one SCF pattern. It also indicates that the entry was rejected by the BHT. The value of :FREQSCORE is 0.25778344, which is larger than the confidence threshold of 0.05. Another, successful lexical entry for *attribute* is shown in figure 2.6. This entry for the SCF 56 49 (e.g. *She attributes her success to hard work*) is large, gathering information from 36 distinct subcategorization patterns.

(22) SCF entry:

```
#S(EPATTERN :TARGET |attribute| :SUBCAT (VSUBCAT NP_PP)
   :CLASSES ((43 44) 2)
   :RELIABILITY 0 :FREQSCORE 0.25778344
   :FREQCNT 1
   :TLTL (VVD)
   :SLTL
   ((|he| PPHS1))
   :OLT1L
   ((|failure| NN1))
   :OLT2L
   ((PSUBCAT SING)
   ((|to| II)) ((|no-one| PN)) ((|buy| VVG)))
   :OLT3L NIL :LRL 0)
```

Legend:

```
#S(EPATTERN :TARGET |verb| :SUBCAT (syntax of arguments for SCF)
   :CLASSES ((SCF number code(s)) frequency of SCF in ANLT)
   :RELIABILITY parse reliability threshold :FREQSCORE score assigned by BHT
   :FREQCNT number of observations in data
   :TLTL  (POS tags for the verb)
   :SLTL  (POS tags for argument heads in subject position)
   :OLT1L (POS tags for argument heads in first argument position)
   :OLT2L (POS tags for argument heads in second argument position)
   :OLT3L (POS tags for argument heads in third argument position)
   :LRL   number of lexical rules applied)
```

In sum, Briscoe and Carroll's approach to acquiring SCFs assumes the following:

- Most sentences will not allow the application of all possible rules of English complementation.
- Some sentences will be unambiguous even given the indeterminacy of the grammar.
- Many incorrect analyses will yield patterns which are unclassifiable and are thus filtered out.
- Arguments of a specific verb will occur with greater frequency than adjuncts in potential argument positions.
- The hypothesis generator will incorrectly output patterns for certain SCF classes more often than others.
- Even a highest ranked pattern for scf_i is only a probabilistic cue to membership of scf_i, so membership should only be inferred if there are enough occurrences of patterns for scf_i in the data to outweigh the error probability for scf_i.

The overall performance of this system was discussed earlier in section 2.5.2, where the system was reported to perform similarly with less ambitious extant systems. The

```
#S(EPATTERN :TARGET |attribute| :SUBCAT (VSUBCAT NP_PP)
                   :CLASSES ((56 49) 2115)
                   :RELIABILITY 0 :FREQSCORE 2.6752692e-25
                   :FREQCNT 36
                   :TLTL
                    (VVZ VVZ VVZ VVZ VVZ VVZ VVZ VVZ VVN VVG VVG VVG VVG VVG
                     VVG VVG VVG VVG VVG VVD VVD VVD VVD VVD VVD VV0 VV0 VV0
                     VV0 VV0 VV0 VV0 VV0 VV0 VV0 VV0)
                   :SLTL
                    (((|text| NN1)) ((|literature| NN1)) ((|he| PPHS1))
                     ((|he| PPHS1)) ((|account| NN1)) ((|He| PPHS1))
                     (((|text| NN1)) ((|literature| NN1)) ((|he| PPHS1))
                     ((|he| PPHS1)) ((|account| NN1)) ((|He| PPHS1))
                     ((|He| PPHS1)) ((|He| PPHS1)) ((|medicine| NN1))
                     ((|what| DDQ)) ((|serve| VV0)) ((|prefer| VV0))
                     ((|laid| VVD)) ((|it| PPH1)) ((|audience| NN))
                     ((|People| NN)) ((|It| PPH1)) ((|Attributing| VVG))
                     ((|Aristotle| NP)) ((|she| PPHS1)) ((|occupation| NN1))
                     ((|institutions| NN2)) ((|government| NN))
                     ((|Prentice| NP)) ((|He| PPHS1)) ((|which| DDQ))
                     ((|study| NN1)) ((|reports| NN2)) ((|one| PN1))
                     ((|it| PPH1)) ((|attribute| VV0)) ((|We| PPIS2))
                     ((|We| PPIS2)) ((|We| PPIS2)) ((|This| DD1))
                     ((|It| PPH1))
                  :OLT1L
                    (((|validity| NN1)) ((|effect| NN1)) ((|ideas| NN2))
                     ((|ideas| NN2)) ((|role| NN1)) ((|this| DD1))
                     ((|success| NN1)) ((|succession| NN1))
                     ((|inferiority| NN1)) ((|content| NN1))
                     ((|characteristics| NN2)) ((|it| PPH1))
                     ((|situation| NN1)) ((|properties| NN2))
                     ((|beliefs| NN2)) ((|disturbances| NN2)) ((|work| NN1))
                     ((|value| NN1)) ((|ideas| NN2)) ((|lack| NN1))
                     ((|it| PPH1)) ((|failure| NN1)) ((|crash| NN1))
                     ((|value| NN1)) ((|ideas| NN2)) ((|lack| NN1))
                     ((|it| PPH1)) ((|failure| NN1)) ((|crash| NN1))
                     ((|win| NN1)) ((|role| NN1)) ((|difficulties| NN2))
                     ((|contribution| NN1)) ((|number| NN1))
                     ((|variability| NN1)) ((|success| NN1)) ((|this| DD1))
                     ((|reality| NN1)) ((|grasp| NN1)) ((|effect| NN1))
                     ((|weight| NN1)) ((|whole| NN1))
                 :OLT2L
                    ((PSUBCAT NP)
                    (((|to| |to| |to| |to| |to| |to| |to| |to| |to| |to| |to| |to| |to|
                       |to| |to| |to| |to| |to| |to| |to| |to| |to| |to| |to| |to| |to|
                       |to| |to| |to| |to| |to| |to| |to| |to| |to| . |to| )II))
                    ((|intention| NN1) (|variables| NN2) (|characters| NN2)
                     (|characters| NN2) (|processes| NN2) (|methods| NN2)
                     (|allusions| NN2) (|sort| NN1) (|being| NN1) (|it| PPH1)
                     (|them| PPHO2) (|childhood| NN1) (|error| NN1)
                     (|systems| NN2) (|ancients| NN2) (|group| NN)
                     (|hand| NN1) (|first| MD) (|thinker| NN1)
                     (|indulgences| NN2) (|machinations| NN2)
                     (|conditions| NN2) (|fault| NN1) (|collapse| NN1)
                     (|vanguard| NN1) (|nature| NN1) (|dismissal| NN1)
                     (|conditions| NN2) (|fault| NN1) (|collapse| NN1)
                     (|vanguard| NN1) (|nature| NN1) (|dismissal| NN1)
                     (|them| PPHO2) (|mind| NN1) (|asset| NN1)
                     (|process| NN1) (|objects| NN2) (|Fido| NP)
                     (|combination| NN1) (|principle| NN1) (|this| DD1)
                  :OLT3L NIL :LRL 0)
```

Figure 2.6: A sample SCF entry

experimental evaluation reported in Briscoe and Carroll (1997) showed that both the hypothesis generation and hypothesis selection phases need refinement. The weakest link in the system proved, however, to be hypothesis selection. The entire approach to filtering needs improvement the better to deal with low frequency SCFs and to yield better overall performance.

2.6 Summary

In this chapter, we have discussed the background and motivation for our work. We first described the phenomenon of verb subcategorization and the account of this phenomenon in linguistic theory, establishing why subcategorization is one of the most important type of information a computational lexicon should provide. We then discussed subcategorization lexicons; the requirements of these resources and attempts to obtain them (semi-)manually. After explaining why (semi-)manual work has not yielded adequate enough lexicons, we argued that automatic subcategorization acquisition is the avenue to pursue.

We surveyed various approaches to automatic subcategorization acquisition. Within a decade, the systems have developed from those capable of learning a small number of SCFs automatically from corpus data, to those capable of detecting a comprehensive set of SCFs and producing large-scale lexicons containing data on the relative frequencies of different SCFs and verb combinations.

Although this is an encouraging development, our review of evaluation indicated that the accuracy of resulting lexicons shows room for improvement. Analysis of error reveals problems common to different systems, arising during hypothesis generation and selection. We pointed out that, while analysis of corpus data has developed significantly during the past decade, the same cannot be said of the filtering methods used for hypothesis selection, which are reported to perform especially poorly. When aiming to improve SCF acquisition, improving hypothesis selection is thus critical. We established this as the scope of our research. We concluded the section by introducing the system employed as framework for SCF acquisition in all the work reported in this thesis.

Chapter 3

Hypothesis Testing for Subcategorization Acquisition

3.1 Introduction

As discussed in chapter 2, nearly all subcategorization acquisition approaches proceed in two steps: generating hypotheses for SCFs and deciding which hypotheses are reliable. The latter step is needed to remove the noise which inevitably arises in SCF acquisition. Most approaches employ statistical hypothesis tests for this purpose (e.g. Brent, 1993; Manning, 1993; Ersan and Charniak, 1996; Lapata, 1999; Briscoe and Carroll, 1997; Sarkar and Zeman, 2000). Despite the popularity of these tests, they have been reported to be inaccurate. As a consequence, hypothesis selection appears to be the weak link in many SCF acquisition systems. The aim of this chapter is to address this problem by examining why hypothesis tests do not perform in SCF acquisition as expected.

In section 3.2, we first provide some theoretical background on hypothesis testing in general. Then in section 3.3, we consider hypothesis testing in the context of subcategorization acquisition, reviewing the tests used and discussing the problems reported with them. In section 3.4, a more detailed examination is provided of the performance of hypothesis testing. We report experiments we conducted to compare three different filtering methods within the framework of Briscoe and Carroll's (1997) SCF acquisition system. Our results show that two hypothesis tests perform poorly, compared with a simple method of filtering SCFs on the basis of their MLEs. We discuss reasons for this, point out a number of problems with hypothesis testing for SCF acquisition, and consider possible directions for further research. Finally, section 3.5 summarises our discussion.

3.2 Background on Hypothesis Testing

Hypothesis testing, as used in SCF acquisition, involves making decisions. In statistics, decision making belongs to the study of inference problems called 'decision theory'.

Generally speaking, decision theory involves formally defining all elements of the decision-making process, including the desired optimality criteria. These criteria are then used to compare alternative decision procedures.

One element of a decision problem is the 'data' described by a random vector $\mathbf{X}$ with sample space $\mathcal{X}$. Another element is a 'model', a set of possible probability distributions for $\mathbf{X}$, indexed by a parameter θ. This parameter is the true but unknown state of nature about which we wish to make an inference. The set of possible values for θ is called the parameter space (Θ). Thus the model is a set $\{f(x|\theta) : \theta \in \Theta\}$ where each $f(x|\theta)$ is a probability mass function or probability density function on $\mathcal{X}$. After the data $\mathbf{X} = x$ is observed, a decision regarding the parameter θ is made. The set of allowable decisions is the 'action space', denoted by ($\mathcal{A}$). The action space determines the type of inference problem with which we are concerned.

When the decision problem is a hypothesis testing problem, the goal is to decide, from a sample of the population, which of the two complementary hypotheses is true: the 'null hypothesis' H_0 or the 'alternative hypothesis' H_1. Hypothesis testing is performed by formulating H_0, which is assumed true unless there is evidence to the contrary. If there is evidence to the contrary, H_0 is rejected and H_1 is accepted. Essentially, a hypothesis test is a rule that specifies

i For which sample values the decision is made to accept H_0 as true

ii For which sample values H_0 is rejected and H_1 is accepted as true

The subset of the sample space for which H_0 will be rejected is called the 'rejection region' or 'critical region'. The complement of the rejection region is called the 'acceptance region'.

Thus in terms of decision theory, only two actions are allowable in hypothesis testing, either "accept H_0" or "reject H_0". When denoting these two actions a_0 and a_1, respectively, the action space in hypothesis testing is the two point set $\mathcal{A} = \{a_0, a_1\}$. A decision rule ($\delta(x)$) is a rule that specifies, for each $x \in \mathcal{X}$, what action $a \in \mathcal{A}$ will be taken if $\mathbf{X} = x$ is observed. In hypothesis testing we have

$$\delta(x) = a_0 \quad \text{for all } x \text{ that are in the acceptance region of the test}$$

$$\delta(x) = a_1 \quad \text{for all } x \text{ that are in the rejection region of the test}$$

In deciding to accept or reject H_0, we may make a mistake. Two types of error may be distinguished:

Type I Error The hypothesis test incorrectly rejects H_0

Type II Error The hypothesis test incorrectly accepts H_0

These two different situations are depicted in figure 3.1. Supposing R denotes the rejection region for a test, the probability of Type I Error is $P(\mathbf{X} \in R|H_0)$ and the probability of the Type II error is $P(\mathbf{X} \in R^C|H_1)$. Hypothesis tests are often

	Accept H_0	Reject H_0
H_0	Correct decision	Type I error
H_1	Type II error	Correct decision

Figure 3.1: Two types of error in hypothesis testing

evaluated and compared through their error probabilities. When doing so, Type II error is frequently minimised subject to a pre-specified value for Type I error. That value is the 'significance' of the test. The significance is often set at 0.05, in which case we have a 'confidence' of 95% in accepting H_0.

The hypothesis tested may refer to a certain parameter of the distribution of the data. For example, we may have a hypothesis about the population mean. Tests of such hypotheses are called 'parametric' tests, and they assume some distribution for the data (e.g. the binomial, normal, t distribution). Examples of parametric tests are the binomial hypothesis test, the log likelihood ratio test and the t test which we shall discuss further in section 3.3. Some tests, on the other hand, are designed for hypotheses about other characteristics of the distribution, such as the similarity between the distributions of two samples. Such tests are called 'non-parametric' (e.g. the Chi-Square test) or 'distribution-free', when they do not assume any distribution for the data (e.g. the Fisher's exact test).

Typically, a parametric hypothesis test is specified in terms of a 'test statistic', a function of the sample $W(\mathbf{X})$. A test might, for example, specify that H_0 is to be rejected if $\overline{X}$, the sample mean, is greater than 3. In this case, $W(\mathbf{X}) = \overline{X}$ is the test statistic and the rejection region is $\{(x_1, ..., x_n) : \overline{x} > 3\}$. Different test statistics (e.g. likelihood ratio tests, invariant tests, Bayesian tests) and rejection regions can be defined. The choice depends upon what sort of departures from the hypothesis we wish to detect[1].

3.3 Hypothesis Testing in Subcategorization Acquisition

When applying hypothesis testing to SCF acquisition, the task is to examine whether, on the basis of accumulated evidence, there is a genuine association between a particular verb ($verb_j$) and a SCF (scf_i). As the input data to statistical filtering is noisy, each occurrence of $verb_j$ has some non-zero probability of being followed by a cue for scf_i, even if it cannot in fact occur with scf_i. The more often $verb_j$ occurs, the more likely it is to occur at least once with a cue for scf_i. Hypothesis testing considers each occurrence of $verb_j$ without a cue for scf_i as a small item of evidence against

[1] For a more detailed account of decision theory and hypothesis testing, see e.g. Casella and Berger (1990) and Kalbfleisch (1985).

$verb_j$ occurring with scf_i. The aim is to determine when $verb_j$ occurs with cues for scf_i often enough to indicate that all those occurrences are unlikely to be errors.

Given this, the null hypothesis H_0 is that there is no association between $verb_j$ and scf_i. Meanwhile, the alternative hypothesis H_1 is that there is such an association. The test is 'one-tailed' since H_1 states the direction of the association, which is a positive correlation between $verb_j$ and scf_i. The expected probability of scf_i occurring with $verb_j$ if H_0 is true is compared with the observed probability of co-occurrence obtained from the corpus data. If the observed probability is greater than the expected probability, we reject H_0 and accept H_1, and if not, we retain H_0.

So far, three hypothesis tests have been used in SCF acquisition: the binomial hypothesis test, the log likelihood ratio test, and the t test. We discuss these tests and their performance in following three sections.

3.3.1 Binomial Hypothesis Test

The most frequently employed statistical test in SCF acquisition is the binomial hypothesis test (BHT), originally introduced for the purpose by Brent (1993) and subsequently used by Manning (1993), Ersan and Charniak (1996), Lapata (1999), Briscoe and Carroll (1997), and Sarkar and Zeman (2000). In section 2.5.3, we introduced Briscoe and Carroll's version of BHT. We shall now look at the test and its different versions in more detail.

Applying this test to SCF acquisition requires recording the total number of SCF cues (n) found for $verb_j$, and the number of these cues for scf_i (m). It also requires an estimate of the error probability (p^e) that a cue for a scf_i occurs with a verb which does not take scf_i. Occurrences of verbs with different putative SCFs are regarded as independent Bernoulli trials. The probability of an event with probability p happening exactly m times out of n such trials is given by the following binomial distribution:

$$P(m, n, p) = \frac{n!}{m!(n-m)!} p^m (1-p)^{n-m} \tag{3.1}$$

The probability of the event happening m or more times is:

$$P(m+, n, p) = \sum_{k=m}^{n} P(k, n, p) \tag{3.2}$$

Finally, $P(m+, n, p^e)$ is the probability that m or more occurrences of cues for scf_i will occur with a verb which is not a member of scf_i, given n occurrences of that verb. A threshold on this probability, $P(m+, n, p^e)$, is usually set at less than or equal to 0.05. This yields a 95% or greater confidence that a high enough proportion of cues for scf_i have been observed for the verb legitimately to be assigned scf_i.

Approaches to SCF acquisition which use a binomial hypothesis test typically differ in respect of the calculation of error probability. Brent (1993) estimates p^e for each SCF experimentally from the behaviour of his SCF extractor. Let N be a lower limit on the number of verb occurrences in the sample. For each scf_i, we can build a histogram where the height of the m^{th} bin is the number of verbs that cue for scf_i exactly m times out of their first N occurrences. Assume that there is some $1 \leq j_0 \leq N$ such that most verbs not taking scf_i are seen with cues for scf_i j_0 times or fewer and, conversely, that most verbs seen with cues for scf_i j_0 times or fewer do not take scf_i. The distribution for $m \leq j_0$ occurrences should be roughly binomial, i.e. proportional to $P(m, N, \hat{p}^e)$, where $\hat{p}^e$ denotes an estimate of p^e for scf_i. Brent's procedure examines each possible estimate j of j_0. For each j, he estimates $\hat{p}^e$ as the average rate among the first N occurrences at which verbs in bins up to j cue for scf_i. The plausibility of j is evaluated by normalizing the first j bins, setting the rest to zero, comparing with $P(m, N, \hat{p}^e)$, and taking the sum of the squared differences between the two distributions. The estimate giving the closest fit between predicted and observed distributions is chosen as the best estimate of p^e.

Brent's calculation of error probability was presumably adopted without changes by Lapata (1999) and Sarkar and Zeman (2000). Manning (1993), however, found that for some SCFs, this method leads to unnecessary low estimates for p^e. Since Brent's cues were sparse but unlikely to be false, the best performance was found with values of the order of 2^{-8}. This was not the case with Manning's approach, where the number of available cues was increased at the expense of reliability of these cues. To maintain high levels of accuracy, Manning applied empirically determined[2] higher bounds on the error probabilities for certain SCFs. The high bound values ranged from 0.25 to 0.02. This approach was also employed by Ersan and Charniak (1996).

When estimating p^e in the manner of Brent or Manning, one makes the assumption that the error probabilities for SCFs are uniform across verbs. This assumption is false, as noted by Brent (1993). Most verbs can, for example, take an NP argument, while very few can take an NP followed by a tensed clause. Assuming uniform error probability results in too few verbs being classified as taking an NP argument and too many taking an NP followed by a tensed clause. This suggests that in calculation of p^e, a better approach would be to take into account variation on the percentage of verbs that can appear in each frame. Briscoe and Carroll (1997) take a step in this direction by estimating p^e as follows:

$$p^e = \left(1 - \frac{|verbs\ in\ scf_i|}{|verbs|}\right) \frac{|patterns\ for\ scf_i|}{|patterns|} \tag{3.3}$$

Briscoe and Carroll extract the number of verb types which are members of the target SCF in the ANLT dictionary. They then convert this number to a probability of frame membership by dividing by the total number of verb types in the dictionary. The complement of this probability provides an estimate for the probability of a verb not taking scf_i. Secondly, this probability is multiplied by an estimate for the probability of observing the cue for scf_i. This is estimated using the number of patterns for i extracted from the SUSANNE corpus, divided by the total number of patterns. According to this estimation, if the probability of observing the cue for scf_i

[2] Manning provides no further details of the empirical estimation.

is 0.5 and the probability of frame membership is only 0.1, the error probability of associating verbs with scf_i is 0.45. However, if the probability of frame membership is 0.5 instead, the error probability is only 0.25.

As shown above, Briscoe and Carroll's estimation of p^e takes into account the relative frequency of verb types that appear in each frame. However, since based on a dictionary, it does not consider the relative frequency of tokens of verb types. It is probable that to obtain more accurate estimates, the number of verb types in ANLT scf_i should be weighted by the frequency of these verbs. It is also possible that the patterns extracted from the SUSANNE corpus are not representative enough to yield fully accurate estimation.

Briscoe, Carroll and Korhonen (1997) apply a method which iteratively optimizes the error probabilities obtained using Briscoe and Carroll's estimation. The idea is similar to that of Manning (1993), i.e. to set high bounds on the error probabilities. The method is based on automatically adjusting the pattern frequencies shown in equation 3.3 on the basis of the errors (false positives and false negatives) the SCF acquisition system makes. First, the errors the system has made are analysed. Then an 'optimal' confidence threshold is calculated such that if the BHT filter had applied it instead of the actual confidence threshold of 0.05, the errors with SCFs would have been minimised. This is done by initially setting a threshold between each pair of SCF occurrences in system output, and then choosing the threshold which yields the minimum number of errors as the optimal confidence threshold. Let p_{scf_i} be the probability assigned to scf_i by the binomial hypothesis test and n the total number of SCFs in system output. The number of incorrect SCFs for each possible threshold is calculated as follows:

$$Errors_i = \begin{cases} \frac{p_{scf_i} + p_{scf_{i+1}}}{2}, & {\scriptstyle i=1,..,n-1} \\ \lim_{j \to i+} p_{scf_j}, & {\scriptstyle i=0} \\ \lim_{j \to i-} p_{scf_j}, & {\scriptstyle i=n} \end{cases} \tag{3.4}$$

The threshold which yields the smallest number of incorrect SCFs is chosen as the optimal threshold p_{opt}. Next, the distance between the optimal threshold and the actual confidence threshold (p_{thr}) is calculated by dividing the latter by the former. The resulting value is multiplied with the pattern frequency of scf_i. This gives an optimised pattern frequency:

$$|patterns\ for\ scf_i|_{opt} = |patterns\ for\ scf_i| \left(\frac{p_{thr}}{p_{opt}} \right) \tag{3.5}$$

Correcting errors in the above manner will generate some new errors. Accordingly, the whole process is repeated until the pattern frequencies and hence the error probabilities are optimised to give the optimum system results with type precision and recall.

Briscoe, Carroll and Korhonen (1997) report an experiment using this method, where the error probabilities were first iteratively optimised with held-out training data covering 10 verbs and then evaluated with test data covering 20 verbs[3]. Using the optimised error probabilities improved SCF acquisition performance by 8.8% with type precision, 20.7% with type recall and 10.9% with ranking accuracy, as compared with the performance with original error probabilities. This result demonstrates that Briscoe and Carroll's estimation of p^e is not optimal. However, the optimization method yields only 70% type precision, 62% type recall and 77% ranking accuracy at its best, which also leaves for room improvement. Closer analysis of results revealed that the method was sufficient only to improving the performance of medium-to-high frequency SCFs.

Nearly all approaches using BHT report that the test is unreliable for SCFs with a frequency of less than 10 (Brent, 1993; Manning, 1993; Ersan and Charniak, 1996; Briscoe and Carroll, 1997). In practice, the poor performance of BHT with low frequency SCFs results in low recall, as many correct SCFs are missed and wrong ones selected.

3.3.2 Log Likelihood Ratio

The binomial log-likelihood ratio (LLR) test (Dunning, 1993) seems theoretically more promising than BHT for low frequency data. The test has been recommended for use in NLP since it does not assume a normal distribution, which invalidates many other parametric tests for use with natural language phenomena. Moreover, it is used in a form ($-2log\lambda$) which is asymptotically χ^2 distributed. This asymptote is appropriate at quite low frequencies, which renders the hypothesis test potentially useful when dealing with natural language phenomena, where low frequency events are commonplace. Dunning (1993) demonstrates the benefits of the LLR statistic in practice, compared with Pearson's chi-squared, on the task of ranking bigram data. The (LLR) test (Dunning, 1993) has been used in SCF acquisition by Sarkar and Zeman (2000) and by Gorrell (1999), who applies it to the SCF acquisition framework of Briscoe and Carroll (1997). Both Sarkar and Zeman and Gorrell use the test in the same way.

To calculate the binomial log-likelihood ratio test, four counts are required for each verb and SCF combination. These are the number of times that:

1. the target verb occurs with the target SCF (k_1)
2. the target verb occurs with any other SCF ($n_1 - k_1$)
3. any other verb occurs with the target SCF (k_2)
4. any other verb occurs with any other SCF ($n_2 - k_2$)

These are the counts from a contingency table, such as that shown below, where the rows indicate the presence or absence of the verb and the columns indicate the presence or absence of the SCF:

[3] The corpus data and method used for evaluation were identical to those used by Briscoe and Carroll (1997), see chapter 2 for details.

	SCF	¬ SCF	Totals
verb	SCF & verb (k_1)	¬ SCF & verb	n_1
¬ verb	SCF & ¬ verb (k_2)	¬ SCF & ¬ verb	n_2

The statistic $-2log\lambda$ is calculated as follows:

$$\begin{aligned} \text{log-likelihood} &= 2[logL(p_1, k_1, n_1) \\ &\quad +logL(p_2, k_2, n_2) \\ &\quad -logL(p, k_1, n_1) \\ &\quad -logL(p, k_2, n_2)\,] \end{aligned} \tag{3.6}$$

where

$$logL(p, n, k) = k \times \log p + (n - k) \times \log(1 - p)$$

and

$$p_1 = \frac{k_1}{n_1},\ p_2 = \frac{k_2}{n_2},\ p = \frac{k_1 + k_2}{n_1 + n_2}$$

The LLR statistic provides a score that reflects the difference in (i) the number of bits it takes to describe the observed data, using $p1 = p(\text{SCF}|verb)$ and $p2 = p(\text{SCF}|\neg verb)$, and (ii) the number of bits it takes to describe the expected data using the probability $p = p(\text{SCF}|any\ verb)$.

The LLR statistic detects differences between $p1$ and $p2$. The difference could potentially be in either direction, but with SCF acquisition, one is interested in LLRs where $p1 > p2$, i.e. where there is a positive association between the SCF and the verb. For these cases, the value of $-2log\lambda$ is compared to the threshold value obtained from Pearson's Chi-Squared table, to see if it is significant at the 95% level.

Surprisingly, both Gorrell (1999) and Sarkar and Zeman (2000) report that with SCF acquisition, LLR yields worse overall performance than BHT. Gorrell reports that when compared to BHT, LLR shows a 12% decline in system performance with type precision, 3% with type recall, and 7% with ranking accuracy. Sarkar and Zeman report a 6% decline in token recall with LLR, as compared with BHT. Gorrell, who provides more detailed analysis of errors, does not find evidence that LLR would even perform better on low frequency classes than BHT.

3.3.3 The *t* test

The *t* test is applied to SCF acquisition only by Sarkar and Zeman (2000). It is derived from the log likelihood ratio test for the normal distribution. Relying on the normal approximation, it is only reliable for large enough SCF samples ($n_i{\cdot}p_i > 5$, $n_i{\cdot}(1{-}p_i) > 5$) and therefore not theoretically as promising for the use of SCF acquisition as LLR. Given a sample from a normal distribution with unknown mean and variance, the test is used to make hypotheses about the mean. It examines the difference between the observed and expected means, scaled by the variance of the data, and indicates how likely it is to get a sample of that (or more extreme) mean and variance, assuming that the sample is drawn from a normal distribution with mean μ.

When applied to SCF acquisition, the value of the t test is used to measure the association between $verb_j$ and scf_i. Using the definitions from section 3.3.2, the test is computed as follows:

$$T = \frac{p_1 - p_2}{\sqrt{\sigma_1(n_1, p_1)^2 + \sigma_2(n_2, p_2)^2}} \tag{3.7}$$

where

$$\sigma(n, p) = \sqrt{\frac{p(1-p)}{n}} \tag{3.8}$$

The value of T has the t distribution with $n_1 + n_2 - 2$ degrees of freedom (which is about normal for large samples). The larger that value is, the more confident we can be that p_1 is greater than p_2 and thus that null hypothesis should be rejected.

Sarkar and Zeman report that the t test performs similarly with LLR, showing only 0.5% improvement over LLR with token recall. No further analysis of errors is provided.

3.4 Comparison of Three Methods

None of the hypothesis tests used in SCF acquisition so far yields accurate enough performance. Although they have been widely reported as problematic, especially with low frequency SCFs, the reasons for poor performance have not been investigated. To examine why these tests perform poorly in SCF acquisition, we performed a series of experiments within the framework of the Briscoe and Carroll's SCF acquisition system[4]. In these experiments, we compared the performance of the Brent style binomial filter of Briscoe and Carroll and the LLR filter of Gorrell (1999) with the performance of a simple method which uses a threshold on the MLEs of SCFs. This section reviews these experiments, discusses the results obtained and considers directions for future work. The three filters are described in section 3.4.1. The details of the experimental evaluation are supplied in section 3.4.2. Our findings are discussed in section 3.4.3 and future work in section 3.4.4.

3.4.1 Methods

When investigating the filtering performance, we used Briscoe and Carroll's SCF system as a framework (see section 2.5.3). In these experiments, the hypothesis generator

[4] The research reported in this section was undertaken in collaboration with Genevieve Gorrell and Diana McCarthy. Full report of our joint work can be found in Korhonen, Gorrell and McCarthy (2000).

of the system (the tagger, lemmatizer, parser, pattern extractor and classifier) was held constant, the only difference being that a parser was used different from that selected by Briscoe and Carroll[5]. While they employed a probabilistic LR parser, our data was parsed using a probabilistic chart parser (PCP) (Chitrao and Grishman, 1990)[6]. Otherwise, the filter was the only component we experimented with. We compared the performance of the system with three different filters:

- The BHT filter of Briscoe and Carroll (1997)
- The LLR filter of Gorrell (1999)
- A new filter which uses a threshold on MLEs of SCFs

The two statistical filters have been described in detail earlier. Section 2.5.3 described the version of BHT used by Briscoe and Carroll, while section 3.3.2 provided details of Gorrell's LLR filter. The new filtering method was applied in order to examine the baseline performance of the system without employing any notion of the significance of the observations. The method involves extracting SCFs classified by the system's classifier, and ranking them in the order of probability of their occurrence with the verb ($p(scf_i|verb_j)$). Probabilities are estimated simply by using a maximum likelihood estimate (MLE) from observed relative frequencies. This is the ratio of count for $scf_i + verb_j$ over the count for $verb_j$. A threshold, determined empirically, is applied to these probability estimates to filter out the low probability entries for each verb. We determined the threshold using held-out training data: such value was chosen which gave optimum average filtering results (according to F measure) for a set of verbs. This yielded a threshold value of 0.02, which was used in the experiments reported below.

3.4.2 Experimental Evaluation

Method

To evaluate the different filters, we took a sample of 10 million words of the BNC corpus. We extracted all sentences containing an occurrence of one of the following fourteen verbs: *ask, begin, believe, cause, expect, find, give, help, like, move, produce, provide, seem, swing.* These verbs originally used by Briscoe and Carroll (1997) were chosen at random, subject to the constraint that they exhibited multiple complementation patterns. After the extraction process, we retained 3000 citations, on average, for each verb. The sentences containing these verbs were processed by the hypothesis generator of the SCF acquisition system, and then the three filtering methods described above were applied. We also obtained results for a baseline without any filtering.

[5] We are indebted to John Carroll for providing us with extracted patterns used in these and other experiments reported in this thesis.

[6] See McCarthy (2001, p. 136) for evaluation of the PCP we employed and the LR parser Briscoe and Carroll (1997) employed. In this evaluation, the LR parser proved slightly more accurate than the PCP, but the differences were not statistically significant.

Method	High Freq			Medium Freq			Low Freq			Totals		
	TP	FP	FN	TP	FP	FN	TP	FP	FN	TP	FP	FN
BHT	75	29	23	11	37	31	4	23	15	90	89	69
LLR	66	30	32	9	52	33	2	23	17	77	105	82
MLE	92	31	6	0	0	42	0	0	19	92	31	67

Table 3.1: Raw results for 14 test verbs

Method	Type Precision %	Type Recall %	F measure
BHT	50.3	56.6	53.3
LLR	42.3	48.4	45.1
MLE	74.8	57.8	65.2
baseline	24.3	83.5	37.6

Table 3.2: Precision, recall and F measure

The results were evaluated against a manual analysis of corpus data, the same manual analysis as employed by Briscoe and Carroll. It was obtained by analysing around 300 occurrences for each of the 14 test verbs in LOB (Garside *et al.*, 1987), SUSANNE and SEC (Taylor and Knowles, 1988) corpora. A manual analysis of the BNC data might produce better results. However, since the BNC is a balanced and heterogeneous corpus, we felt it was reasonable to test the data on a different corpus which is also balanced and heterogeneous.

Following Briscoe and Carroll (1997), we calculated type precision (percentage of SCFs acquired which were also exemplified in the manual analysis) and type recall (percentage of the SCFs exemplified in the manual analysis which were acquired automatically). We also combined precision and recall into a single measure of overall performance using the F measure.

Results

Table 3.1 gives the raw results for the 14 verbs using each method. It shows the number of true positives (TP), false positives (FP), and false negatives (FN), as determined by manual analysis. The results for high frequency SCFs (above 0.01 relative frequency), medium frequency (between 0.01 and 0.001) and low frequency (below 0.001) SCFs are listed respectively in the second, third and fourth columns. These three frequency ranges were defined so that a roughly similar number of SCFs would occur in each range. The final column includes the total results for all frequency ranges.

Table 3.2 shows type precision, type recall and the F measure for the 14 verbs. We also provide the baseline results, if all SCFs were accepted.

From the results given in tables 3.1 and 3.2, it is apparent that the MLE approach outperformed both hypothesis tests. For both BHT and LLR there was an increase in FNs at high frequencies, and an increase in FPs at medium and low frequencies, when compared with MLE. The number of errors was typically larger for LLR than BHT. The hypothesis tests reduced the number of FNs at medium and low frequencies, but this

Method	**Type Precision %**	**Type Recall %**	**F measure**
BHT	62.5	55.1	58.6
LLR	50.9	47.0	48.9

Table 3.3: Results with small BNC data

was countered by the substantial increase in FPs that they gave. While BHT nearly always acquired the three most frequent SCFs of verbs correctly, LLR tended to reject these.

While the high number of FNs can be explained by reports which have shown LLR to be over-conservative (Ribas, 1995; Pedersen, 1996), the high number of FPs is surprising. Although theoretically the strength of LLR lies in its suitability for low frequency data, the results displayed in table 3.1 do not suggest that the method performs better than BHT on low frequency frames.

MLE thresholding produced better results than the two statistical tests used. Precision improved considerably, showing that SCFs occurring in the data with highest frequency are often correct. Also recall showed slight improvement as compared with BHT and LLR. Although MLE thresholding clearly makes no attempt to solve the sparse data problem, it performs better than BHT or LLR overall. MLE is not adept at finding low frequency SCFs: the other methods are, however, problematic in that they wrongly accept more than they correctly reject. The baseline, of accepting all SCFs, obtained a high recall at the expense of precision. It performed 7.5 worse according to the F measure than LLR, showing that even a poor filtering method yields better overall performance than no filtering at all.

Interestingly, we have some further results which suggest that both BHT and LLR perform better when less data is used. When we run the same experiment using only an average of 1000 citations of each verb from the sample of 10 million words of the BNC, precision and recall are improved, as seen in table 3.3. This is surprising since statistical tests take sample size into account and should be more reliable as the sample size increases. We performed cross-validation which confirmed that this effect holds across different subsets of BNC. Each of the three subsets examined showed better performance with smaller sample.

3.4.3 Discussion

Our results indicate that MLE outperforms both hypothesis tests. We found two explanations for this, which we believe are jointly responsible.

Firstly, the SCF distribution is approximately zipfian, as are many distributions concerned with natural language (Manning and Schütze, 1999). In a zipf-like distribution, the product of rank order (r) and frequency (f) is constant. According to Zipf's law:

$$f \propto \frac{1}{r} \tag{3.9}$$

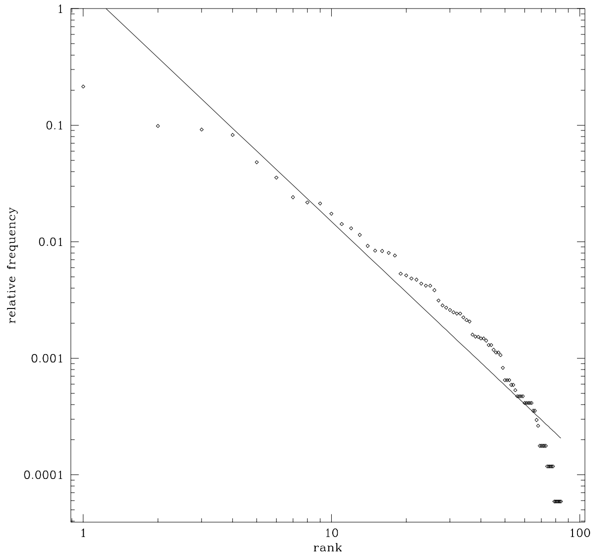

Figure 3.2: Hypothesised SCF distribution for *find*

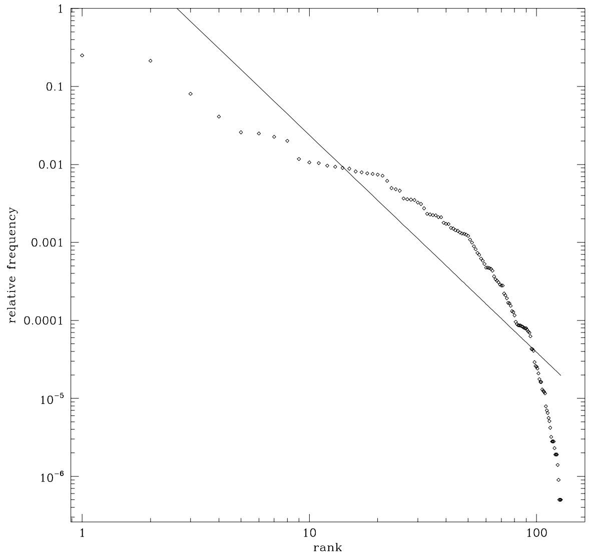

Figure 3.3: Hypothesised unconditional SCF distribution

In other words, there is a constant k such that $f \cdot r = k$.

Figures 3.2 and 3.3 display two zipf plots. The former shows the conditional SCF distribution for the verb *find*, while the latter shows the unconditional distribution of SCFs for all verbs. These unfiltered SCF probability distributions were obtained by running the pattern classifier of Briscoe and Carroll's system on 20 million words of BNC. The figures show SCF rank on the X-axis versus SCF relative frequency on the Y-axis, using logarithmic scales. The line indicates the closest Zipf-like power law fit to the data. These figures illustrate typical zipfian skewed distributions where the few very high frequency SCFs have several orders of magnitude more occurrences than most others. There is a middling number of medium frequency SCFs and a long tail of low frequency SCFs.

Secondly, the hypothesis tests make the false assumption (H_0) that the unconditional and conditional distributions are correlated. The fact that a significant improvement in performance is made by optimizing the prior probabilities for SCFs according to the performance of the system (Briscoe, Carroll and Korhonen, 1997; see section 3.3.1) suggests the discrepancy between unconditional and conditional distributions.

We examined the correlation between the manual analysis for the 14 verbs and the unconditional distribution of verb types over all SCFs estimated from ANLT using the Kullback-Leibler Distance (KL) and Spearman Rank Correlation Coefficient (RC). The results included in table 3.4 show that the distributions compared are fairly dissimilar and that only a moderate to poor rank correlation was found averaged over all verb types[7]. Manual inspection of SCFs taken by individual verbs shows that this result is not surprising. For example, the highest ranked SCF type with the verb *believe* is a sentential complement. This SCF type is not as common, however, with verbs in general, ranked only as 12th among the SCF types in ANLT. Furthermore, while the MLE for sentential complement is 0.48 with *believe*, it is only 0.012 with verb types in general.

Both LLR and BHT work by comparing the observed value of $p(scf_i|verb_j)$ with that expected by chance. They both use the observed value for $p(scf_i|verb_j)$ from the system's output, and they both use an estimate for the unconditional probability distribution ($p(scf)$) for estimating expected probability. They differ in the way in which the estimate for unconditional probability is obtained and the way that it is used in hypothesis testing.

For BHT, the null hypothesis is that the observed value of $p(scf_i|verb_j)$ arose by chance, because of noise in the data. We estimate the probability that the value observed could have arisen by chance using $p(m+, n, p^e)$. p^e is calculated using:

- the SCF acquisition system's raw (unfiltered) estimate for the unconditional distribution, which is obtained from the SUSANNE corpus and
- the ANLT estimate of the unconditional distribution of a verb not taking scf_i, across all SCFs

[7] Note that KL ≥ 0, with KL near to 0 denoting strong association, and $-1 \leq$ RC ≤ 1, with RC near to 0 denoting a low degree of association and RC near to -1 and 1 denoting strong association. See section 2.5.2 for full account of both KL and RC.

Verb	KL	RC
ask	1.25	0.10
begin	2.55	0.83
believe	1.94	0.77
cause	0.85	0.19
expect	1.76	0.45
find	1.29	0.33
give	2.28	0.06
help	1.59	0.43
like	1.39	0.56
move	0.78	0.53
produce	0.53	0.95
provide	0.44	0.65
seem	3.32	0.16
swing	0.79	0.50
Average	1.48	0.47

Table 3.4: Kullback-Leibler distance and Spearman rank correlation between the conditional SCF distributions of the test verbs and unconditional distribution

For LLR, both conditional (p_1) and unconditional (p_2) estimates are obtained from the BNC data. The unconditional probability distribution uses the occurrence of scf_i with any verb other than our target.

The binomial tests look at one point in the SCF distribution at a time, for a given verb. The expected value is determined using the unconditional distribution, on the assumption that if the null hypothesis is true then this distribution will correlate with the conditional distribution. However, this is rarely the case. Moreover, given the zipfian nature of the distributions, the frequency differences at any point can be substantial. In these experiments, we used one-tailed tests because we were looking for cases where there was a positive association between the SCF and verb, however, in a two-tailed test the null hypothesis would rarely be accepted, because of the substantial differences in the conditional and unconditional distributions.

A large number of false negatives occurred for high frequency SCFs because the probability with which we compared them was too high. This probability was estimated from the combination of many verbs genuinely occurring with the frame in question, rather than from an estimate of background noise from verbs which did not occur with the frame. We did not use an estimate from verbs which do not take the SCF, since this would require a priori knowledge about the phenomena that we were endeavouring to acquire automatically. For LLR the unconditional probability estimate (p_2) was high, simply because this SCF was common, rather than because the data was particularly noisy. For BHT, p^e was likewise too high as the SCF was also common in the SUSANNE data. The ANLT estimate went some way to compensating for this; thus we obtained fewer false negatives with BHT than LLR.

A large number of false positives occurred for low frequency SCFs because the estimate for $p(scf_i)$ was low. This estimate was more readily exceeded by the conditional estimate. For BHT false positives arose because of the low estimate of $p(scf_i)$ (from

SUSANNE) and because the estimate of $p(\neg scf_i)$ from ANLT did not compensate enough for this. For LLR, there was no means to compensate for the fact that p_2 was lower than p_1.

In contrast, MLE did not compare two distributions. Simply rejecting the low frequency data produced better results overall by avoiding false positives with the low frequency data, and false negatives with the high frequency data.

3.4.4 Conclusion

Further work on handling low frequency data in SCF acquisition is warranted. With hypothesis tests, one possibility is to put more effort into estimation of p^e, and to avoid use of the unconditional distribution for this. For example, Manning and Schütze (1999) propose supplementing BHT with prior knowledge about a verb's SCFs. This could be done by stipulating a higher prior for SCFs listed for a verb in some dictionary. In some further experiments with BHT, we optimised the estimates for p^e depending on the performance of the system for the target SCF, using the method proposed by Briscoe, Carroll and Korhonen (1997) (see section 3.3.1). The estimates of p^e were obtained from a held-out training set separate from the BNC data used for testing. Results using the new estimates for p^e showed no improvement with low frequency SCFs. They gave an overall improvement of 10% type precision and 6% type recall, compared to the BHT results reported here[8]. Nevertheless, the result was 14% worse for precision than MLE, though there was a 4% improvement in recall, making the overall performance 3.9 worse than MLE according to the F measure.

Methods based on optimising estimates for p^e are likely to represent an upper bound to BHT's accuracy. BHT and other hypothesis tests applied in SCF acquisition so far assume that the different SCFs taken by $verb_j$ occur independently. Several researchers have questioned this assumption (Carroll and Rooth, 1998; Manning and Schütze, 1999; Sarkar and Zeman, 2000). Manning and Schütze (1999) and Sarkar and Zeman (2000) propose modeling the dependence between different SCFs for $verb_j$ using a multinomial distribution. To our knowledge this method has yet not been tried. While we agree that the independence assumption is arguably questionable, it is unclear how this method would address the problems we have identified with BHT and LLR.

A non-parametric or distribution-free statistical test, such as Fisher's exact test recommended by Pedersen (1996), might improve on the results obtained using parametric tests. The computation for this test, however, can quickly become cumbersome as a calculation is required for every possible configuration of the contigency table that results in the observed marginal totals. Moreover, Pedersen's results did not appear to demonstrate a significant advantage compared with LLR. On the task of identifying bigrams, the ranks assigned by the LLR and Fisher's exact test are identical.

As known from other areas of NLP, the zipfian nature of data alone remains a chal-

[8] This improvement obtained using the optimization method is smaller than that reported in section 3.3.1. This is due to the optimization method's being dependent on the accuracy of the baseline results. As the baseline BHT results reported in this section were not as accurate as those reported in section 3.3.1 (e.g. due to the differences in test data) the improvement gained was smaller.

lenge for both parametric and non-parametric statistical tests[9]. The frequent and infrequent ranges of a zipfian distribution exhibit very different statistical behaviour. It is possible that no statistic can be found that would work well for both high and medium-to-low frequency events and thus allow direct comparison of the significance of both rare and common phenomena. Also, as Briscoe (2001) points out, zipfian data is by nature inadequate from the statistical learning point of view, regardless of the amount and accuracy of the data used. Because the power law is scaling invariant, no finite sample will be representative in the statistical sense. In addition, power law distributions often indicate that we sample from a non-stationary rather than a stationary source (Casti, 1994). This partly explains why statistical models of learning, which rely on representative samples from stationary sources, do not perform optimally.

The better result obtained using MLE is to some extent supported by Lapata (1999) who reported that a threshold on the relative frequencies produced slightly better results than those achieved with a Brent-style binomial filter when establishing SCFs for diathesis alternation detection. However, Lapata's approach differs from ours in that she determined thresholds for each SCF (independently from verbs) using the frequency of the SCF in BNC and COMLEX. The method fails to account for the fact that SCF probabilities are not uniform across the verbs. Better results would be obtained if the variation on the percentage of tokens of verb types that can appear in each frame was taken into account.

To improve the performance of MLE, it would be worth investigating ways of handling low frequency data for integration with this method. Any statistical test would work better at low frequencies than the MLE, since this simply disregards all low frequency SCFs. If in our experiments, we had used MLE only for high frequency data, and BHT for medium and low, then overall we should have had 54% precision and 67% recall. For integration with MLE, it seems worth employing hypothesis tests which do not rely on the unconditional distribution for low frequency SCFs. Another option would be to integrate MLE with smoothing. This approach would avoid altogether the use of statistical tests. However, more sophisticated smoothing methods, which back-off to an unconditional distribution, will also suffer from the lack of correlation between conditional and unconditional SCF distributions. In other words, only if the unconditional SCF distribution provided accurate back-off estimates for SCFs, could it be used to smooth the conditional distributions to compensate for the poor performance on rare SCFs and to detect SCFs unseen.

3.5 Summary

In this chapter, we have discussed the problem of statistical filtering in subcategorization acquisition. After providing theoretical background on the theory of hypothesis testing, we reviewed hypothesis tests applied in SCF acquisition and described the problems associated with them. We then reported experiments with Briscoe and Car-

[9] For example, Manning and Schütze (1999) discuss the performance of various hypothesis tests on the task of identifying collocations, and Kilgarriff (2001) evaluates different statistical tests used for comparing corpora. They both report poor performance with these tests on zipfian data.

roll's SCF acquisition system, where we explored three possibilities for filtering SCF entries produced by the system. These were (i) a version of the binomial hypothesis test filter, (ii) a version of the binomial log-likelihood ratio test filter and (iii) a simple method using a threshold on the MLEs of the SCFs hypothesised. Surprisingly, the simple MLE thresholding method worked best. The BHT and LLR both produced an astounding number of FPs, particularly at low frequencies. Our investigation showed that hypothesis testing does not work well because not only is the underlying distribution zipfian but also there is very little correlation between conditional and unconditional SCF distributions. BHT and LLR wrongly assumed such a correlation for H_0 and thus were susceptible to error. The lack of correlation between the conditional and unconditional SCF distributions will, however, also affect refinements of MLE such as smoothing or Bayesian estimation. Sophisticated methods for handling sparse data would benefit from more accurate back-off estimates for SCFs than the unconditional SCF distribution can provide.

Chapter 4

Back-off Estimates for Subcategorization Acquisition

4.1 Introduction

In chapter 3, we discussed the poor performance of statistical tests frequently employed for hypothesis selection in SCF acquisition. Our investigation showed that one substantial source of error lies in the lack of accurate back-off estimates[1] for SCFs, delimiting the filtering performance. However, access to more accurate back-off estimates would not only benefit widely-used statistical filters, but any method employed for hypothesis selection which relies on such estimates. It would also help e.g. the simple filter based on MLE thresholding (introduced in chapter 3) which requires refinement the better to deal with sparse data.

In this chapter we shall consider ways of obtaining more accurate back-off estimates for SCF acquisition. The poor correlation between the unconditional ($p(scf)$) and conditional SCF distributions ($p(scf|verb)$) suggests that no single set of back-off estimates is applicable to all verbs. Rather, it is likely that verbs of different subcategorization behaviour require different estimates. In the following sections we consider linguistic resources which classify verbs according to their distinctive subcategorization behaviour. We examine whether back-off estimates could be based on the verb classes these resources provide ($p(scf|class)$).

We start by introducing the linguistic verb classifications we plan to explore (section 4.2). We then report experiments where we compare how well verbs grouped similarly in these classifications correlate in terms of SCF distributions (section 4.3). The outcome from these experiments is summarized in section 4.4[2].

[1] We use the term 'back-off estimates' in a broad sense to refer to the SCF probability estimates used for guiding SCF acquisition is some way. We make no reference to the particular method employed (e.g. hypothesis testing, smoothing, Bayesian estimation, etc.).

[2] See Korhonen (2000) for a summary of the central experimental findings presented this and the following chapter.

4.2 Methods of Verb Classification

In the previous chapter, poor correlation was reported between unconditional ($p(scf)$) and conditional SCF distributions ($p(scf|verb)$). Unlike approaches to SCF acquisition have so far generally assumed, $p(scf)$ does not provide accurate back-off estimates for $p(scf|verb)$. This is not actually surprising, considering that individual verbs differ largely in terms of the number and type of SCFs they take. For instance, a verb like *ignore* takes only one SCF (NP), while a verb like *believe* takes multiple SCFs (e.g. NP, PART-NP, NP-PP, PP, PART-NP-PP, INTRANS, PART, NP-ADJP, NP-PP-PP). Given this, a single set of back-off estimates is unlikely to fully account for the SCF variations the different verbs pose. Instead, it is likely that verbs of different subcategorization behaviour require different back-off estimates. A verb like *clip*, for instance, which intuitively takes near identical set of SCFs with *cut* (e.g. NP, PART-NP, NP-PP, PP, PART-NP-PP, INTRANS, PART, PART-PP, NP-PP-PP) should require similar back-off estimates to *cut*.

An alternative is thus to classify verbs into classes distinctive in terms of subcategorization and obtain back-off estimates specific to these classes ($p(scf|class)$). Some lexical resources exist which associate verbs with classes that capture subcategorization behaviour characteristic to their members. These classifications have been obtained on both semantic and syntactic grounds. In the following, we shall first describe the semantically and then the syntactically-driven verb classifications used in our work.

4.2.1 Semantic Verb Classification

Two current approaches to semantically-driven verb classification, both widely used within NLP research, are the Levin classes (Levin, 1993) and WordNet (Miller *et al.*, 1993). Levin's taxonomy of verbs and their classes is based on diathesis alternations. Verbs which display the same alternations in the realization of their argument structure are assumed to share certain meaning components and are organized into a semantically coherent class. WordNet, on the other hand, is a semantic network based on paradigmatic relations which structure the different senses of verbs. Of the two sources, Levin classes are more interesting to us, since they provide sets of SCFs associated with individual classes. WordNet classifies verbs on a purely semantic basis without regard to their syntactic properties. Although the syntactic regularities studied by Levin are to some extent reflected by semantic relatedness as it is represented by WordNet's particular structure (Dorr, 1997; Fellbaum, 1999), WordNet's semantic organization does not always go hand in hand with a syntactic organization. Levin classes thus give us a better starting point. WordNet provides, however, useful information not included in Levin classes, for example, information about different semantic relations between verbs and the frequency of verb senses. Unlike Levin's taxonomy, it is also a comprehensive lexical database. We thus use WordNet as a source of additional information. We shall next introduce these two classifications in more detail.

Levin's Semantic Verb Classes

Levin verb classes (Levin, 1993) are based on the ability of a verb to occur in specific diathesis alternations, i.e. specific pairs of syntactic frames which are assumed to be meaning retentive. Levin's central thesis is that "the behaviour of a verb, particularly with respect to the expression and interpretation of its arguments, is to a large extent determined by its meaning" (Levin, 1993, p. 1). Thus, according to Levin, the semantics of a verb and its syntactic behaviour are predictably related. The syntactic frames are understood as a direct reflection of the underlying semantic components that constrain allowable arguments. For instance, (23) exemplifies the substance/source alternation. Verbs undergoing this alternation express substance emission. They take two arguments, which Levin characterizes as (i) a source (emitter) (e.g. *sun*) and (ii) the substance emitted from this source (e.g. *heat*). The semantic role of the subject of the intransitive use (23a) of the verb is the same as the semantic role of the object of the transitive use (23b). Similarly the semantic roles of the oblique object of the intransitive use and the subject of the transitive use match.

(23) a **Heat** *radiates from* **the sun**
b **The sun** *radiates* **heat**

Drawing on previous research on diathesis alternations (e.g. Jackendoff, 1990; Pinker, 1989) and her own investigations, Levin defines 79 alternations for English. These alternations concern changes in verbs' transitivity or within the arguments of VP, or involve the introduction of oblique complements, reflexives, passives, *there*-insertion, different forms of inversions or specific words. They are mainly restricted to verbs taking NP and PP complements.

Levin analyses over 3200 verbs according to alternations, associating each verb with the alternation(s) it undergoes. She argues that verbs which behave similarly with respect to alternations share certain meaning component(s), and can thus be grouped together to form a semantically coherent class. Levin puts this idea into practice by proposing 48 semantically motivated classes of verbs whose members pattern in the same way with respect to diathesis alternations and other properties:

> "The classificatory distinctions... involve the expression of arguments of verbs, including alternate expressions of arguments and special interpretations associated with particular expressions of arguments of the type that are characteristic of diathesis alternations. Certain morphological properties of verbs, such as the existence of various types of related nominals and adjectives, have been used as well, since they are also tied to the argument-taking properties of verbs". (Levin, 1993, p. 17)

Some of the classes split further into more distinctive subclasses, making the total number of classes 191. For each verb class, Levin provides key syntactic and semantic characteristics. She does not provide in-depth analysis of meaning components involved in various classes, nor does she attempt to formulate verb semantic representation of the type discussed e.g. with most linking approaches in section 2.3.1.

Rather, her aim is to better set the stage for such future work. According to Levin, examination of classes of verbs defined by shared behaviour can play an important role in identification of meaning components.

Let us consider, as an example, the broad Levin class of "Verbs of Change of State". This class divides into six different subclasses, each of which relates to changes of state in distinguishing ways. For instance, "*Break* Verbs" refer to actions that bring about a change in the material integrity of some entity, while "*Bend* Verbs" relate to a change in the shape of an entity that does not disrupt its material integrity. Each subclass is characterized by its participation or non-participation in specific alternations and/or constructions. "*Break* Verbs" (e.g. *break, chip, fracture, rip, smash, split, tear*) are characterized by six alternations, three of which they permit (24a-c) and three of which they do not permit (24d-f), and by further constructions, as shown in (24g-i).

(24) a **Causative/inchoative alternation**:
Tony broke the window ↔ *The window broke*

b **Middle alternation**:
Tony broke the window ↔ *The window broke easily*

c **Instrument subject alternation**:
Tony broke the window with the hammer ↔ *The hammer broke the window*

d * ***With/against*** **alternation**:
Tony broke the cup against the wall ↔ **Tony broke the wall with the cup*

e ***Conative alternation**:
Tony broke the window ↔ **Tony broke at the window*

f ***Body-Part possessor ascension alternation**:
**Tony broke herself on the arm* ↔ *Tony broke her arm*

g **Unintentional interpretation available (some verbs)**:
Reflexive object: **Tony broke himself*
Body-part object: *Tony broke his finger*

h **Resultative phrase**:
Tony broke the piggy bank open, *Tony broke the glass to pieces*

i **Zero-related Nominal**:
a break, *a break in the window*, **the break of a window*

Membership in specific alternations and constructions yields the syntactic description of this verb class. For instance, the specification in (24) shows that the "*Break* Verbs" take (at least) the INTRANS, NP, NP-PP and NP-ADJP frames.

Levin classification is not exhaustive in terms of breadth or depth of coverage. More work is needed to cover a larger set of diathesis alternations and further to extend and refine verb classification. Also, as Levin mentions, there is a sense in which the whole notion of a verb class is artificial. As most verbs are characterized by several meaning components, there is potential for cross-classification, which in turn means that other, equally viable classification schemes can be identified instead of that proposed. Nevertheless, the current source is unique in providing useful core sets of verbs with specific sets of properties and in being extensive enough for practical NLP

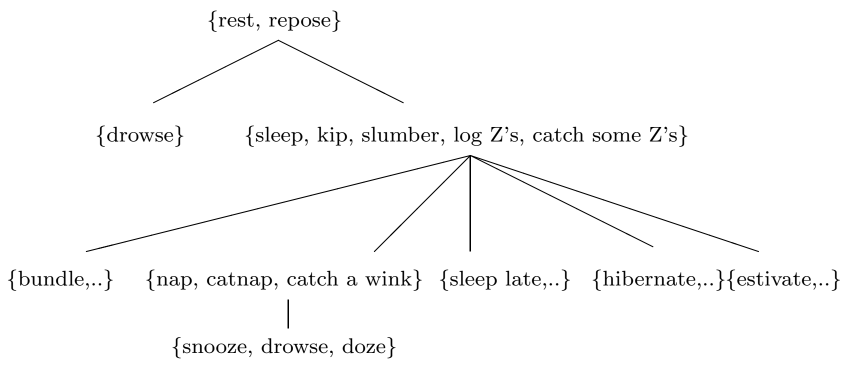

Figure 4.1: A WordNet hierarchy fragment: troponymy relations

use. The particular interest to us is that it links the syntax and semantics of verbs, providing semantically motivated sets of SCFs associated with individual classes.

WordNet

WordNet (Miller *et al.*, 1993) (version 1.6) is an online lexical database of English containing over 120,000 concepts expressed by nouns, verbs, adjectives, and adverbs. In contrast to Levin's classification, WordNet organizes words on a purely semantic basis without regard to their SCFs. As a semantic source, it concentrates on paradigmatic relations and whole lexical items rather than atomic meaning units. The design of WordNet is inspired by psycholinguistic and computational theories of human lexical memory. Its organization is that of a network of interlinked nodes representing word meanings. The nodes are sets of unordered synonym sets ('synsets'), which consist of all the word forms that can express a given concept. For example, the synset containing the verb forms *put, place, set, pose, position* and *lay* stands for the concept, which can be referred to by any one of its members. The members of a synset are not absolute but rough synonyms, so that they can be substituted for each other in most but not all contexts. Word forms and synsets are linked to one another by means of lexical and conceptual-semantic relations. While synonymy links individual words within synsets, the super-/subordinate relation (e.g. 'troponymy' relation with verbs) links entire synsets. The latter relation builds hierarchical structures linking generic to more specific concepts.

We used the verb hierarchy of WordNet version 1.6. It contains 10,319 distinct word forms whose 22,066 senses are organized into 12,127 synsets, representing an equal number of distinct verb meanings. The verb hierarchy consists of 15 mostly semantically-driven subhierarchies each of which accommodates appropriate synsets: "verbs of motion", "perception", "contact", "communication", "competition", "change", "cognition", "consumption", "creation", "emotion", "possession", "stative", "weather", "bodily care and functions" and "social behaviour and interaction". Most verb synsets in these hierarchies are interrelated by a pointer standing for a manner relation troponymy. For example, the synset {*snooze, drowse, doze*} belonging to the subhierarchy of "verbs of bodily care and functions" is represented as one of the troponyms (subordinates) of the hypernym (superordinate) synset {*rest, repose*}, since *snooze, drowse* or *doze* mean to *rest* or *repose* in a particular manner. Figure 4.1 illustrates the part of the WordNet verb hierarchy where these synsets appear among other synsets arranged according to troponymy. Other conceptual-semantic relations linking both entire synsets and individual verb forms are cause, entailment and semantic opposition.

Each verb synset contains, besides all the word forms that can refer to a given concept, a definitional gloss, and - in most cases - an example sentence. Using WordNet search facilities, one can obtain further information about a single word form, e.g. about its different WordNet senses and their frequencies, its synonym(s), hypernym(s), troponym(s), antonym(s) and so forth. Some information about SCFs is also available (for example, basic information about transitivity and argument type) but this information is neither comprehensive nor detailed. The four senses of *lend* e.g. are described in WordNet with the following sentence frames:

(25) Somebody is — ing PP
Somebody — something
Something — something
Somebody — something to somebody
Somebody — somebody PP
Somebody — something PP

These translate into SCFs NP, NP-PP, NP-TO-NP and PP, while it is known that *lend* can also take (at least) the SCFs NP-NP, INTRANS, PART-NP, PART-PP and PP-PP.

4.2.2 Syntactic Verb Classification

A possible source of syntactic verb classification is a large syntax dictionary. Verbs encoded with similar SCF possibilities in a comprehensive dictionary may be assumed to demonstrate similar syntactic behaviour. Verbs can thus be grouped into syntactically coherent classes according to the particular sets of SCFs assigned to them. This approach was previously taken e.g. by Carter (1989). He introduced a lexical acquisition tool for the SRI Core Language Engine (CLE), which allows the creation of CLE lexicon entries using templates based on expected sets of SCFs already exemplified in the CLE lexicon. Carter calls such sets of SCFs ‘paradigms’. He defines a paradigm as “any maximal set of categories (i.e. SCFs) with the same distribution among (lexical) entries” (Carter, 1989, p. 4). We adopt here Carter’s term and definition of a paradigm.

To obtain syntactic verb classification based on paradigms, we used the ANLT dictionary. (Boguraev *et al.*, 1987). ANLT includes 63,000 lexicon entries in total, 23,273 of which are verbal entries. A verbal entry comprises a certain verb form and subcategorization combination. Separate argument structures are thus listed in separate entries, as illustrated by *appear* in figure 4.2. The treatment of subcategorization is fairly thorough, with phrasal, prepositional and phrasal-prepositional verbs also encoded with subcategorization possibilities. Control is encoded and the distinction

```
(appear ”” (V (AGR IT) (ARITY 1) (LAT -) (SUBCAT SFIN)) APPEAR ())
(appear ”” (V (AGR IT) (ARITY 1) (LAT -) (SUBCAT WHS) (SUBTYPE ASIF)) APPEAR ())
(appear ”” (V (AGR IT) (ARITY 2) (LAT -) (PFORM TO) (SUBCAT PP_SFIN)) APPEAR ())
(appear ”” (V (ARITY 1) (LAT -) (SUBCAT NULL)) APPEAR ())
(appear ”” (V (ARITY 1) (LAT -) (SUBCAT SC_AP) (SUBTYPE RAIS)) APPEAR ())
(appear ”” (V (ARITY 1) (LAT -) (SUBCAT SC_INF) (SUBTYPE RAIS)) APPEAR ())
(appear ”” (V (ARITY 1) (LAT -) (SUBCAT SC_NP) (SUBTYPE RAIS)) APPEAR ())
(appear ”” (V (ARITY 2) (LAT -) (PFORM BEFORE) (SUBCAT PP)) APPEAR ())
(appear ”” (V (ARITY 2) (LAT -) (PFORM FOR) (SUBCAT PP)) APPEAR ())
(appear ”” (V (ARITY 2) (LAT -) (PFORM TO) (SUBCAT SC_PP_INF) (SUBTYPE RAIS)) APPEAR ())
(appear ”” (V (ARITY 2) (LAT -) (SUBCAT LOC)) APPEAR ())
```

Figure 4.2: ANLT lexical entries for *appear*

made between object and subject control, as well as equi and raising. Some alternations are included, such as the dative alternation.

ANLT defines subcategorization using feature value pairs. The main features are SUBCAT which describes the arguments a verb subcategorizes for; SUBTYPE, which provides further information about a particular subcategorization; ARITY, which lists the number of logical arguments; and PFORM and PRT, which indicate subcategorizations concerning prepositions and particles of a particular type. More or less specific SCF classifications can be obtained, depending on which features and values are taken to be distinctive. Briscoe's (2000) SCF classification in Appendix A, for example, includes 127 SCF distinctions from ANLT. The SCFs abstract over specific lexically-governed particles and prepositions, but make use of most other distinctions provided in ANLT[3].

The lexical entry in (26) e.g. could be used to describe subcategorization in *He appears crazy.* In this case, *appear* is a subject control verb which subcategorizes for adjectival phrase (SUBCAT SC_AP). It is also a raising verb (SUBTYPE RAIS) of latinate origin (LAT -), and takes only one logical argument (ARITY 1).

(26) (appear ”” (V (ARITY 1) (LAT -) (SUBCAT SC_AP) (SUBTYPE RAIS)) APPEAR)

We extracted from ANLT all possible paradigms, i.e. all different sets of SCFs associated with verbs, assuming the classification of 127 SCFs by Briscoe (2000). 742 different paradigms were identified, ranging from those including only one SCF (e.g. NP) to those including over 30 SCFs. After this, all verbs in ANLT were grouped according to the paradigms they take. Each verb was classified as a member of one paradigm only, that whose SCFs include all and only the SCFs associated with the verb in ANLT. Relative frequency for different paradigms was also calculated, based on the number of ANLT verb types associated with them. (27) exemplifies an ANLT paradigm which comprises six SCFs[4]:

[3] See Briscoe (2000) for further details.

[4] Instead of using the ANLT feature value pairs to indicate different SCFs, we use here simple abbreviations from the SCF classification included in Appendix A. See this classification for the mapping between the abbreviations and the ANLT feature value pairs.

(27) a INTRANS: *The ship loaded*
b NP: *They loaded the ship*
c NP-PP : *They loaded the ship with oranges*
d PART: *The ship loaded up*
e PART-NP: *They loaded up the ship*
f PART-NP-PP: *They loaded up the ship with oranges*

This particular paradigm is associated in ANLT with six different verbs: *drum, flood, flush, load, marry* and *shut.* Thus according to ANLT, these verbs exhibit identical syntactic behaviour. (28) shows the entry for this paradigm in our syntactic verb classification.

(28) ANLT paradigm: INTRANS, NP, NP-PP, PART, PART-NP, PART-NP-PP
Relative frequency: 0.0009
Members: *drum, flood, flush, load, marry, shut*

4.3 Experiments with Subcategorization Distributions

We conducted experiments to investigate whether the verb classifications introduced above are in practice distinctive enough in terms of subcategorization to provide an adequate basis for back-off estimates. This was done by examining the degree of correlation between conditional SCF distributions for individual verbs classified similarly in these resources. Section 4.3.1 gives details of the SCF distributions used in these experiments and section 4.3.2 describes the measures used for examining the degree of correlation between the distributions. The experiments with semantically similar verbs are reported in section 4.3.3, and those with syntactically similar verbs in section 4.3.4.

4.3.1 SCF Distributions

We used two methods for obtaining the SCF distributions used in our experiments. The first was to acquire an unfiltered subcategorization lexicon for 20 million words of BNC using Briscoe and Carroll's system. This gives us the "observed" distribution of SCFs for individual verbs and that for all verbs in the BNC data. The second method was manually to analyse around 300 occurrences[5] of each individual verb examined in the BNC data. The SCFs were analysed according to Briscoe and Carroll's classification (Appendix A). This gives us an estimate of the "correct" SCF distributions for the individual verbs. The estimate for the correct distribution of SCFs over all English verbs was obtained by extracting the number of verbs which are members of each SCF in the ANLT dictionary. In this, we assumed Briscoe's (2000) definition of an ANLT SCF.

[5] Manual analysis of around 300 occurrences was discovered by Briscoe and Carroll (1997) sufficient to obtain an adequate SCF distribution for gold standard.

Both the observed and correct estimates are specific to verb form rather than sense. The observed estimates for SCFs are noisy, but they are all reported over verb tokens. The correct estimates are more accurate, but only those for individual verbs are reported over verb tokens. The correct estimates for all English verbs are over verb types since, due to the lack of comprehensive manual analysis for all English verbs, they were obtained from the ANLT dictionary. As neither the observed nor the correct estimates are ideal, we used both in our experiments to verify that the results obtained with one generalize to the other.

4.3.2 Measuring Similarity between Distributions

The degree of SCF correlation was examined by calculating the Kullback-Leibler distance (KL) and the Spearman rank correlation coefficient (RC) between the different distributions. The details of these measures were were given in section 2.5.2. Let us recall now that while KL measures the dissimilarity between the distributions ($\text{KL} \geq 0$, with KL near to 0 denoting strong association), RC measures the similarity in ranking of SCFs between the distributions ($-1 \leq \text{RC} \leq 1$, with RC near to 0 denoting a low degree of association and RC near to -1 and 1 denoting strong association).

4.3.3 SCF Correlation between Semantically Similar Verbs

To examine the degree of SCF correlation between semantically similar verbs, we took Levin's verb classification as a starting point. Levin classes are based on associations between specific SCFs and verb *senses*. However, subcategorization acquisition systems are so far capable of associating SCFs with verb *forms* only, as no WSD is employed. Thus while Levin has shown that verbs from the same semantic class are similar in terms of verb sense specific subcategorization, our aim was to investigate whether verbs from the same class are also similar in terms of verb form specific subcategorization. In addition, we are not only interested in intersections of SCFs between verbs but also in the degree of correlation between SCF distributions and in the ranking of SCFs in these distributions. The Levin classes nevertheless provide us with a useful starting point. We examined (i) to what extent verbs from the same Levin class correlate in terms of SCF distributions specific to verb form and (ii) whether the factors of sense frequency (i.e. predominant vs. non-predominant sense), polysemy (i.e. the number of senses taken by a verb), semantic relations (i.e. hypernym vs. hyponym[6]), and the specificity of Levin class assumed (broad class vs. subclass) affect this correlation.

Focusing on five broad Levin classes - "Verbs of Change of Possession", "Assessment Verbs", "Verbs of Killing", "Verbs of Motion", and "*Destroy* Verbs" - we chose five test verbs from each class. These verbs were chosen so that one is a generic hypernym of the other four verbs. We used WordNet for defining and recognising this semantic relation. We defined a hypernym as a test verb's hypernym in WordNet, and a hyponym as a verb which, in WordNet, shares this same hypernym with a test verb.

[6] We do not differentiate between hyponymy and troponymy relations but for the rest of this thesis use the term hyponym to refer either to troponym or hyponym.

13. Verbs of Change of Possession	Test Verbs	No. of WN Senses
13.1 *Give* Verbs	**give**	45
13.2 *Contribute* Verbs	*contribute*	4
13.3 Verbs of Future Having	*offer*	13
13.4 Verbs of Providing	*provide*	4
13.6 Verbs of Exchange	change	10
43. Verbs of Assessment	Test Verbs	No. of WN Senses
	analyse	3
	explore	4
	investigate	2
	survey	6
	observe	9
42. Verbs of Killing	Test Verbs	No. of WN Senses
42.1 *Murder* Verbs	**kill**	14
42.1 *Murder* Verbs	*murder*	2
42.1 *Murder* Verbs	*slaughter*	2
42.2 *Poison* Verbs	*strangle*	1
42.1 *Murder* Verbs	execute	7
44. Destroy Verbs	Test Verbs	No. of WN Senses
	destroy	4
	ruin	2
	demolish	2
	waste	1
	devastate	7
51. Verbs of Motion	Test Verbs	No. of WN Senses
51.3 Manner of Motion verbs	**move**	16
51.1 Verbs of Inherently Directed Motion	*arrive*	2
51.4 Verbs of Motion Using a Vehicle	*fly*	14
51.4 Verbs of Motion Using a Vehicle	*sail*	4
51.2 *Leave* Verbs	abandon	2

Table 4.1: Levin test verbs

Three of the four hyponyms were required to have their predominant sense involved in the Levin class examined, while one of them was required to have its predominant sense in some other verb class. Predominant sense was defined by manually examining the most frequent sense of a verb in WordNet and by comparing this with the Levin sense in question[7].

Table 4.1 shows the test verbs employed. The first column lists the number and name of each broad Levin class and specifies the possible Levin subclass an individual verb belongs to[8]. The second column lists, for each Levin class, the five individual test verbs. The hypernym verb for the other verbs in the same class is indicated in bold font. The three hyponyms whose predominant sense is involved with the Levin class in question are indicated in italic font. The one hyponym whose predominant

[7] We acknowledge that WordNet sense frequency information was obtained from the Brown corpus and therefore cannot be taken as definite but rather instructive.

[8] We only consider subclasses, ignoring possible further divisions into sub-subclasses. Where no subclass is given, the broad Levin class does not divide further.

sense is not involved with the Levin class in question is listed last, using normal font. The third and final column shows the number of senses assigned to each test verb in WordNet. This indicates the degree of polysemy.

For instance, table 4.1 lists five "Change of Possession" verbs: *give, contribute, offer, provide* and *change*. The hypernym of the four other verbs is *give. Contribute, offer, provide* and *change* are its hyponyms; the predominant sense of *change*, however, is not with this verb class (rather, with the Levin "Verbs of Change of State"). The class of "Change of Possession" verbs consists of several subclasses. The five verbs each belong to a different subclass. The degree of polysemy between these verbs varies largely. The hypernym *give* e.g. is highly polysemic with 45 distinct senses in WordNet, while *contribute* and *provide* each have only four WordNet senses.

All other test verbs are listed in Levin (1993) with the verb class indicated in this table, except *explore, investigate, survey* and *observe*, which are listed with "*Investigate* Verbs". We re-assigned these verbs to "Assessment Verbs", since they also fulfil the characteristics of that class. In addition, their predominant sense in WordNet is associated with "Assessment", rather than with "*Investigate* Verbs".

In these experiments, we took from each verb class the three hyponyms whose predominant sense belongs to the verb class in question and examined the degree to which the SCF distribution for each of these verbs correlates with the SCF distributions for three other verbs from the same Levin class. The latter verbs were chosen so that one is the hypernym of a test verb, while the two others are hyponyms - one with predominant sense in the relevant verb class and the other with some other verb class. For comparison, we also examined how well the SCF distribution for the different test verbs correlates with the SCF distribution of all English verbs in general and with that of a semantically different verb (i.e. a verb belonging to a different Levin class).

The results given in tables 4.2 and 4.3 were obtained by correlating the "observed" SCF distributions from the BNC data. Table 4.2 shows an example of correlating the SCF distribution of the "Motion" verb *fly* against that of (i) its hypernym *move*, (ii) hyponym *sail*, (iii) hyponym *abandon*, whose predominant sense is not with motion verbs, (iv) all verbs in general, and (v) *agree*, which is not related semantically. The results show that the SCF distribution for *fly* clearly correlates more closely with the SCF distribution for *move*, *sail* and *abandon* than that for all verbs and *agree*.

The average results for all test verbs given in table 4.3 indicate that, according to both KL and RC, the degree of SCF correlation is closest with semantically similar verbs. Hypernym and hyponym relations are nearly as good, the majority of verbs showing slightly better SCF correlation with hypernyms. As one might expect, sense frequency affects the degree of SCF correlation. Of the two hyponym groups, that whose predominant sense is involved with the Levin class examined show closer SCF correlation. The correlation between individual verbs and verbs in general, is poor, but still better than with semantically unrelated verbs.

These findings for observed SCF distributions hold as well for "correct" SCF distributions, as seen in tables 4.4 and 4.5. The average results given in table 4.4 are closely similar to those given in table 4.3. Table 4.5 shows that in terms of SCF distributions, verbs in all classes examined correlate more closely with their hypernym verbs than

		KL	RC
fly	*move*	0.25	0.83
fly	*sail*	0.62	0.61
fly	*abandon*	0.82	0.59
fly	all verbs	2.13	0.51
fly	*agree*	2.27	0.12

Table 4.2: Correlating the SCF distribution of *fly* against other SCF distributions

	KL	RC
hypernym	0.65	0.71
hyponym (predominant sense)	0.71	0.66
hyponym (non-predominant)	1.07	0.63
all verbs	1.59	0.41
semantically different verb	1.74	0.38

Table 4.3: Overall correlation results with observed distributions

	KL	RC
hypernym	0.44	0.66
hyponym (predominant sense)	0.76	0.59
hyponym (non-predominant)	0.89	0.54
all verbs	1.19	0.39
semantically different verb	1.62	0.27

Table 4.4: Overall correlation results with correct distributions

	Hypernym		**All Verbs**	
Verb Class	KL	RC	KL	RC
change of possession	0.61	0.64	1.16	0.38
assessment	0.28	0.71	0.73	0.48
killing	0.70	0.63	1.14	0.37
destroy	0.30	0.60	1.19	0.29
motion	0.29	0.73	1.72	0.42
AVERAGE	0.44	0.66	1.19	0.39

Table 4.5: Correlation results for five verb classes with correct distributions

with all verbs in general. However, there are differences between the verb classes such that verbs in one class show closer SCF correlation with the hypernym verb than those in another class. According to our results, these differences are not attributable to the degree of polysemy. The highly polysemic "Change of Possession" verbs, for instance, show the second poorest correlation (among the five verb classes), while the closest occurs between "Verbs of Motion" which are also fairly polysemic. "*Destroy* Verbs" which all have under 10 WordNet senses, show average results. Detailed comparison of results for individual verbs supports these observations. It seems that the degree of polysemy does not affect the SCF correlation as much as sense frequency (i.e. the predominant sense).

Similarly, the specificity of the Levin class assumed does not seem to affect the results. "Motion" verbs examined are mostly from different subclasses, but they still correlate more closely with their hypernym verb than "*Destroy* Verbs", which are all from the same broad class, which does not divide into subclasses. Arguably, there should be better SCF correlation between verbs from an indivisible verb class. On the other hand, Levin (1993) produced her classification according to verb sense, while we extracted SCF distributions specific to verb form. Our polysemic distributions involve a wider range of SCFs than Levin's single sense classes. In addition, Levin's classification is not fully comprehensive and several verb classes require further work before sufficiently clear distinctions can be made.

Overall, these results show that verbs from the same Levin class correlate more closely with other verbs from the same class (especially when classified semantically according to their predominant sense) than with all verbs in general or with semantically different verbs.

4.3.4 SCF Correlation between Syntactically Similar Verbs

To investigate the degree of SCF correlation between syntactically similar verbs, we examined the extent to which verbs taking the same paradigm in the ANLT dictionary correlate in terms of SCF distributions.

In these experiments, we focused on four different ANLT paradigms. The first consists simply of an NP frame. This is the most frequent paradigm in ANLT, with roughly 30% of verb types taking only an NP frame. The second paradigm comprises INTRANS and NP frames. This is the second most common paradigm in ANLT, taken by 12% of the verb types. The third paradigm comprises INTRANS and PP frames, and the fourth INTRANS, NP, PP, and NP-PP frames. These two paradigms are less frequent, with 4% of ANLT verb types taking the former, and 1.7% the latter.

From each of the four paradigms, we chose four ANLT verbs as our test verbs. To ensure that we examine purely syntactic similarity, we required that the verbs from the same paradigm are semantically different. This was verified by manually checking that their senses belong to different Levin classes. Table 4.6 shows the test verbs employed. The first column specifies the verbal paradigm in question and below it the four individual test verbs. The second column lists, for each verb, the ANLT SCFs included in our correct SCF distributions obtained from the manual analysis of BNC data.

ANLT Paradigm: NP	Paradigm from Corpus
acquire	NP, NP-PP
analyse	NP, NP-PP, INTRANS, NP-AS-NP, WH-S
complete	NP, NP-PP, PP
ignore	NP

ANLT Paradigm: INTRANS, NP	Paradigm from Corpus
destroy	INTRANS, NP
hide	INTRANS, NP, PP, NP-PP, ADJP, ADVP, NP-ADVP, PART PART-NP, PART-NP-PP, PART-PP
produce	NP, NP-PP, NP-PP-PRED, NP-TO-INF-SC
slide	INTRANS, NP, PP, NP-PP, NP-ADVP, PART, PART-NP, PART-PP, PP-PP, NP-PART-PP, NP-ADL

ANLT Paradigm: INTRANS, PP	Paradigm from Corpus
arise	INTRANS, PP
arrive	INTRANS, PP, NP-PP, ADVP, PART-PP, PP-PP
differ	INTRANS, PP, P-WH-S
react	INTRANS, PP, NP-PP, ADVP, PP-PP, P-ING-SC, ADVP-PP

ANLT Paradigm: INTRANS, NP, PP, NP-PP	Paradigm from Corpus
remove	INTRANS, NP, PP, NP-PP, NP-PP-PP, P-POSSING
contribute	INTRANS, NP, PP, NP-TO-NP
distinguish	NP, PP, NP-PP, NP-PP-PP, NP-AS-NP, WH-S
visit	INTRANS, NP, PP, NP-PP

Table 4.6: Paradigm test verbs

	Paradigm		**All Verbs**	
ANLT **Paradigm**	KL	RC	KL	RC
NP	0.40	0.80	0.61	0.75
INTRANS, NP	0.76	0.50	0.88	0.48
INTRANS, PP	0.79	0.41	1.34	0.37
INTRANS, NP, PP, NP-PP	1.21	0.31	1.10	0.29
AVERAGE	0.79	0.51	0.98	0.47

Table 4.7: Correlation results for syntactically similar verbs

According to ANLT, the verbs *acquire, analyse, complete* and *ignore* take only an NP frame. Manual analysis of a corpus reveals, however, that all these verbs, except *ignore*, also permit additional SCFs. For instance, *analyse* can take also NP-PP (*We analysed words into phonemes*), INTRANS (*He analysed and analysed*), NP-AS-NP (*Bill analysed the words as nouns*), and WH-S (*John analysed what had gone wrong*) frames.

By just manually comparing the ANLT and corpus paradigms shown in table 4.6 we can see that the paradigms provided by ANLT are not comprehensive. This is due to the nature of static dictionaries: they tend to have high type precision but disappointing type recall. Only 3 of the 16 test verbs do not occur in corpus data with a SCF assigned to them by ANLT. Only 4 of the verbs occurred in corpus data with a paradigm identical with that predicted by ANLT. As many as 12 take additional SCFs not predicted by ANLT, 4 per verb on average.

To examine the SCF correlation, we took from each ANLT paradigm the four individual test verbs and examined the degree to which the SCF distribution for each of these verbs correlates with those for all the other verbs taking the same paradigm. Thus for each paradigm, six pairs of SCF distributions were compared. For comparison, we also examined how well the SCF distribution for the different test verbs correlates with the SCF distribution of all English verbs in general.

We obtained the results given in table 4.7 by correlating the observed SCF distributions from the BNC data. The average results for all paradigms show that, according to both KL and RC, the degree of SCF correlation is closer with syntactically similar verbs than with all verbs in general. However, this difference is smaller than with the semantically similar verbs, especially with rank correlation. Nor does it apply to all the paradigms examined. The verbs taking the ANLT paradigm INTRANS, NP, PP, NP-PP show poorer correlation with each other than with all verbs in general. From these results, it would seem that verbs from a more frequent paradigm, which contains fewer SCFs, show closer mutual SCF correlation. This effect is, however, partly due to our evaluation. The fewer SCFs there are to consider, the less noise enters the evaluation. For this reason we did not use the correct distributions in these experiments. The correct distributions contain fewer SCFs than the observed distributions. All verbs tested for semantic similarity took enough SCFs to yield an adequate test using these distributions, but this was not the case with all the verbs tested for syntactic similarity.

Since, as noted above, dictionaries tend to have low type recall, more comprehensive verbal paradigms could be obtained by combining the syntactic information in several dictionaries. However, it is unclear whether this would give better results for SCF correlation. In our experiment, the test verbs which, according to manual analysis of corpus data, took near-identical sets of SCFs (e.g. *remove* and *distinguish*) did not show noticeably better SCF correlation than the other test verbs examined.

4.4 Summary

In this chapter, we have addressed the problem that the unconditional SCF distribution provides poor back-off estimates for SCF acquisition. We investigated whether more accurate back-off estimates could be obtained by basing them on linguistically-driven verb classes. Employing Levin's semantic verb classification and the syntactic classification obtained from the ANLT dictionary, we examined whether verbs classified similarly in these resources correlate well in terms of their verb form specific SCF distributions. The results showed that the degree of SCF correlation was closer with semantically and syntactically similar verbs than with all verbs in general, and that the correlation between semantically similar verbs was better than that between syntactically similar verbs. The closest SCF correlation was observed when verbs were classified semantically according to their predominant sense. These results suggest that more accurate back-off estimates may be obtained for SCF acquisition by classifying verbs semantically according to their predominant sense and obtaining estimates specific to semantic classes.

Chapter 5

A New Approach to Hypothesis Selection

5.1 Introduction

The experiments reported in chapter 4 suggest that more accurate back-off estimates could be obtained for SCF acquisition by basing them on semantic verb classes. In this chapter, we propose a method for constructing semantically motivated back-off estimates (section 5.2). In addition, we propose a new method for hypothesis selection, which makes use of these estimates (section 5.3). This involves combining the MLE thresholding and smoothing with back-off estimates, allowing us to avoid any problems based on hypothesis testing. To evaluate the back-off estimates and the method for hypothesis selection, we report a series of experiments to examine whether this approach can, in practice, improve the accuracy of subcategorization acquisition (section 5.4). Finally, we consider further work (section 5.5) and summarise discussion (section 5.6).

5.2 Back-off Estimates

In chapter 4, fairly close SCF correlation was reported between pairs of semantically similar verbs. A simple way of obtaining back-off estimates would be to select a single verb from a semantic class and use its conditional SCF distribution as estimates for the other verbs in the same verb class. We propose another method, however, which involves taking the conditional SCF distributions of a few verbs in the same class and merging them to obtain the back-off estimates for the class ($p(scf|class)$). Using several conditional SCF distributions - as opposed to only one - helps to minimise the problem of sparse data and cover the SCF variations within verb classes and variations due to polysemy.

Our method involves constructing back-off estimates specific to broad Levin classes. First, 4-5 representative verbs are chosen from a class, subject to the following constraints, which we verify manually:

1. To reduce the effect of sense frequency, the predominant WordNet sense of each verb must correspond to the Levin class in question.

2. To obtain representative estimates, when possible, the verbs should represent different Levin subclasses.

3. To make use of the benefit that verbs correlate well with their hypernym verb, when possible, one of the verbs should be a hypernym of the other verbs in WordNet.

For the verbs chosen, we obtain correct SCF distributions by manually analysing around 300 occurrences of each verb in the BNC data. The SCFs are analysed according to Briscoe's classification (Appendix A). Finally, the resulting SCF distributions are merged automatically to construct the back-off estimates for the verb class.

Using this method, we obtained the back-off estimates for the Levin class of "Verbs of Motion", for example, by choosing five representative verbs from this class - *march, move, fly, slide* and *sail* - and by merging the SCF distributions of these verbs. Table 5.1 shows, in the first five columns, the SCFs and their relative frequencies for the five individual "Motion" verbs. The SCFs are indicated using the number codes of the SCF classification. They are listed in order of their relative frequency, starting from the highest ranked SCF (e.g. 87 for *slide*) and ending in the lowest ranked SCF(s) (e.g. 24, 120, 155, and 3 for *slide*). The sixth column shows the back-off estimates for the class of "Motion" verbs, obtained by merging the conditional distributions shown in the first five columns. The benefits of using several conditional SCF distributions for obtaining the back-off estimates are visible in this table. The back-off estimates include a wider range of SCFs than any of the conditional distributions alone and embody the average ranking of SCFs, given the five conditional distributions.

5.3 Hypothesis Selection

In chapter 3, a simple method was proposed for filtering SCFs on the basis of their MLEs (MLE thresholding; see section 3.4.1). Experiments were reported which showed that this method outperforms two statistical tests frequently employed for hypothesis selection in SCF acquisition. Given the poor performance of the statistical tests and the problems related to them[1], we decided not to pursue them further. Instead, we chose to refine MLE thresholding. Although the method shows good performance with high frequency SCFs, it requires augmentation the better to deal with low frequency SCFs. A way of addressing this problem is to smooth the MLEs.

Smoothing is frequently used in NLP to deal with problems of sparse data, caused by the inherent zipfian nature of language. It addresses the problem that even for a very large data collection, ML estimation does not allow us adequately to estimate probabilities of rare but nevertheless possible events. Smoothing enables the detection of these events by assigning them some non-zero probability. It does this by making the probability distribution "closer" to some other probability distribution.

[1] Recall the discussion in sections 3.4.3 and 3.4.4.

slide		*fly*		*march*		*sail*		*move*		**Verbs of Motion**	
SCF	r.f.	SCF	r.f.	SCF	r.f	SCF	r.f	SCF	r.f	SCF	b. estimates
87	0.297	22	0.286	87	0.415	87	0.388	22	0.300	87	0.303
74	0.222	87	0.236	22	0.193	22	0.355	87	0.180	22	0.240
76	0.157	24	0.187	76	0.111	24	0.080	74	0.127	74	0.113
49	0.093	78	0.074	78	0.104	74	0.064	24	0.106	24	0.081
78	0.083	74	0.064	74	0.089	78	0.048	119	0.067	76	0.073
22	0.065	49	0.044	95	0.037	76	0.048	78	0.049	78	0.072
95	0.046	3	0.039	24	0.022	95	0.016	3	0.047	49	0.035
24	0.009	95	0.029	49	0.015			95	0.042	95	0.034
120	0.009	76	0.020	27	0.014			76	0.028	3	0.019
155	0.009	160	0.020					49	0.022	119	0.013
3	0.009							77	0.017	160	0.004
								122	0.007	27	0.003
								27	0.007	77	0.003
										120	0.002
										155	0.002
										122	0.001

Table 5.1: SCFs and their relative frequencies in (i) conditional distributions for five individual "Motion" verbs and in (ii) back-off estimates for "Verbs of Motion"

Most smoothing methods work by discounting probability estimates given by ML estimation applied to the observed frequencies and redistributing the freed probability mass among the events which never (or rarely) occurred in data. This can be done simply by assigning a uniform prior to all events, or by employing a more sophisticated method, such as backing-off. The latter method estimates the probability of unseen or low frequency events by backing-off to another probability distribution. Instead of employing discounting, some smoothing methods work by simply combining multiple probability estimates. For example, one can make a linear combination of two probability distributions in the hope of producing a better overall model. Various smoothing techniques have been proposed and applied in the field of NLP. Comprehensive reviews of these techniques can be found e.g. in Jurafsky and Martin (2000) and Manning and Schütze (1999).

Integrating the MLE method with a sophisticated smoothing method allows us to use the semantically motivated back-off estimates before filtering. Specifically, it allows us to classify verbs according to their semantic class and smooth the conditional SCF distributions for these verbs using the back-off estimates of the respective verb class. The following two sections describe how this is done. Section 5.3.1 provides details of the method adopted for hypothesis selection and section 5.3.2 introduces the smoothing methods employed.

5.3.1 Procedure for Hypothesis Selection

The method adopted for hypothesis selection is essentially MLE thresholding (section 3.4.1), with the additional step of smoothing added. It involves extracting the SCFs classified by the classifier of Briscoe and Carroll's system, and ranking them in order of probability of their occurrence with the verb ($p(scf_i|verb_j)$). Probabilities are estimated by using a MLE from the observed relative frequencies, which is the ratio of count for $scf_i + verb_j$ over the count for $verb_j$. The resulting conditional SCF distribution for a verb is then smoothed before filtering the SCFs, using the back-off estimates of the semantic class to which the verb belongs. The details of the smoothing algorithms employed are provided in section 5.3.2. After smoothing, filtering is performed by applying a threshold to the resulting set of probability estimates. Held-out training data is used to establish an optimal threshold for each semantic verb class examined[2]. For each class, such a threshold value is chosen which maximises the average SCF filtering performance (according to F measure) for verbs in a class. A threshold is established on smoothed estimates i.e. it is determined specific to a smoothing method.

5.3.2 Smoothing Methods

Three different smoothing methods were integrated with the overall procedure described above: add-one, Katz backing-off and linear interpolation.

Add One Smoothing

Add-one smoothing (Laplance, 1995) has the effect of giving some of the probability space to the SCFs unseen in the conditional distribution. Unlike the two other smoothing methods employed, it makes no use of back-off estimates. Rather, it provides a baseline smoothing method against which the more sophisticated methods can be compared. Let $c(scf_i)$ be the frequency of a SCF given a verb, N the total number of SCF tokens for this verb in the conditional distribution, and C the total number of SCF types. The estimated probability of the SCF is:

$$P(scf_i) = \frac{c(scf_i) + 1}{N + C} \tag{5.1}$$

Instead of assigning each unseen SCF a frequency of 1, any other small value (λ) could in principle be used. We used held-out training data to confirm that $\lambda = 1$ achieves optimal average smoothing results for test verbs: the SCF distributions obtained using this value correlate best on average with the corresponding gold standard distributions[3] (according to both KL and RC).

Katz Backing-off

Katz backing-off (Katz, 1987) uses back-off estimates. It gives some of the probability space to the SCFs unseen or of low frequency in the conditional distribution. It does

[2] Verb class specific thresholds were used as they gave better results than a uniform threshold. This is not surprising since verb classes differ with respect to the number of SCFs typically taken by their member verbs.

[3] See section 5.4.2 for the gold standard employed.

this by backing-off to another distribution. Let $p_1(scf_i)$ be a probability of a SCF in the observed distribution, and $p_2(scf_i)$ its probability in the back-off distribution. The estimated probability of the SCF is calculated as follows:

$$P(scf_i) = \begin{cases} (1-d) \cdot p_1(scf_i) & if\ c(scf_i) > c_1 \\ \alpha\ \cdot p_2(scf_i) & otherwise \end{cases} \tag{5.2}$$

The cut off frequency c_1 is an empirically defined threshold determining whether to back-off or not. When counts are lower than c_1 they are held too low to give an accurate estimate, and we back-off to a second distribution. In this case, we discount $p_1(scf_i)$ a certain amount to reserve some of the probability space for unseen and very low frequency SCFs. The discount (d) is defined empirically, and α is a normalization constant which ensures that the probabilities of the resulting distribution sum to 1. Held-out training data was used to determine optimal values for both c_1 and d. These values were determined specific to a verb class. Such values were chosen for a class as yield SCF distributions which correlate the most closely with the corresponding gold standard distributions (according to both KL and RC) for member verbs on average.

Linear Interpolation

Like Katz-backing off, linear interpolation (Chen and Goodman, 1996) makes use of back-off estimates. While Katz backing-off consults different estimates depending on their specificity, linear interpolation makes a linear combination of them. The method is used here for the simple task of combining a conditional with the back-off distribution. The estimated probability of the SCF is given by

$$P(scf_i) = \lambda_1(p_1(scf_i)) + \lambda_2(p_2(scf_i)) \tag{5.3}$$

where the λ_j denotes weights for the different distributions and sum to 1. The value for λ_j was obtained by optimising the smoothing performance on the held-out training data for all scf_i. It was determined specific to a verb class, by choosing the value that yields the optimal average smoothing performance for verbs in a class[4]. This was determined by comparing the correlation (according to KL and RC) between SCF distributions obtained using different values for λ_j to corresponding gold standard distributions.

5.4 Experiments

To evaluate the back-off estimates and the new approach of hypothesis selection, we performed experiments which we report below. Section 5.4.1 introduces the test

[4] Note that in all experiments reported in this thesis, the minimum value for λ_2 was set to 0.2. As we used linear interpolation for examining the accuracy of back-off estimates, this minimum value allowed us to examine whether (inaccurate) back-off estimates can also decrease performance.

data and the back-off estimates used in these experiments. Section 5.4.2 describes the evaluation method adopted. Section 5.4.3 reviews evaluation of smoothing, while evaluation of back-off estimates is reported in section 5.4.4.

5.4.1 Data and Back-off Estimates

Test data consisted of a total of 60 verbs from 11 broad Levin classes, listed in table 5.2. One Levin class was collapsed together with another similar Levin class ("Verbs of Sending and Carrying" and "Verbs of Exerting Force"), making the total number of verb classes 10. The test verbs were chosen at random, subject to the constraint that they occurred frequently enough in corpus data[5] and that their most frequent sense in WordNet belonged to the Levin class in question. All the other test verbs, except three "Assessment Verbs" (*explore, investigate, survey*) were listed by Levin (1993) as members of the respective verb class.

From each class, 4-5 suitable test verbs were chosen by hand to construct the back-off estimates for the class. These verbs are indicated in table 5.2 using normal font. Each test verb used in obtaining the estimates was excluded when testing the verb itself. For example, when testing the "Motion" verb *travel*, we used the back-off estimates constructed from the verbs *march, move, fly, slide* and *sail*. When testing *fly*, however, we used the back-off estimates constructed from the verbs *march, move, slide* and *sail* only.

5.4.2 Method of Evaluation

We took a sample of 20 million words of the BNC for evaluation and extracted all sentences containing an occurrence of one of the 60 test verbs, a maximum of 3000 citations of each. The sentences containing these verbs were processed by the SCF acquisition system. The hypothesis generator of the system was held constant, the exception being that the data for these experiments were parsed using a PCP (Chitrao and Grishman, 1990). For hypothesis selection, we employed the new method which applied the different smoothing methods before filtering. We also obtained results for the baseline MLE thresholding method without any smoothing.

The results were evaluated against a manual analysis of the corpus data. This was obtained by analysing a maximum of 300 occurrences for each test verb in the BNC corpora. We calculated type precision, type recall, F measure and ranking accuracy. In addition to the system results, we calculated KL and RC between the acquired unfiltered SCF distributions and the distributions obtained from manual analysis. We also recorded the total number of SCFs unseen in acquired unfiltered SCF distributions which occurred in gold standard distributions. This was to investigate how well the approach tackles the sparse data problem, i.e. the extent to which it is capable of detecting the SCFs altogether missing in the data output by the hypothesis generator.

[5] This restriction was set merely for test purposes. As we evaluated our results against manual analysis of corpus data, we required at least 300 occurrences for each verb to guarantee sufficiently accurate evaluation.

9. Verbs of Putting	
9.1 *Put* Verbs	place
9.2 Verbs of Putting in a Spatial Configuration	lay
9.4 Verbs of Putting with a Specified Direction	drop
9.5 *Pour* Verbs	*pour*
9.7 *Spray/Load* Verbs	load
9.8 *Fill* Verbs	*fill*
11. Verbs of Sending and Carrying, 12. Verbs of Exerting Force	
11.1 *Send* Verbs	send, ship, *transport*
11.3 *Bring* and *Take*	*bring*
11.4 *Carry* Verbs	carry
12. Verbs of Exerting Force	*pull*, push
13. Verbs of Change of Possession	
13.1 *Give* Verbs	give, lend
13.2 *Contribute* Verbs	contribute, donate
13.3 Verbs of Future Having	offer
13.4 Verbs of Providing	*provide, supply*
13.5 Verbs of Obtaining	*acquire, buy*
34. Verbs of Assessment	
	analyse, explore, investigate, survey
36. Verbs of Social Interaction	
36.1 *Correspond* Verbs	*agree*, communicate, *struggle*
36.2 *Marry* Verbs	marry
36.3 *Meet* Verbs	meet, visit
42. Verbs of Killing	
42.1 *Murder* Verbs	kill, murder, slaughter
42.2 *Poison* Verbs	strangle
44. *Destroy* Verbs	
	demolish, destroy, ruin, devastate
48. Verbs of Appearance, Disappearance and Occurrence	
48.1 Verbs of Appearance	arise, emerge
48.2 Verbs of Disappearance	disappear, vanish
51. Verbs of Motion	
51.1 Verbs of Inherently Directed Motion	*arrive, depart*
51.3 Manner of Motion Verbs	march, move, slide, *swing, travel, walk*
51.4 Verbs of Motion Using a Vehicle	fly, sail
51.5 *Walz* Verbs	*dance*
55. Aspectual Verbs	
55.1 *Begin* Verbs	begin, start
55.2 *Complete* verbs	end, complete, terminate

Table 5.2: Test data

			System results				Unseen
Method	KL	RC	Rank A. (%)	Precision (%)	Recall (%)	F	SCFs
Baseline	1.41	0.50	33.3	60.0	33.3	42.8	4
Add-one	1.67	0.27	33.3	60.0	33.3	42.8	0
Katz b.	1.58	0.58	33.3	60.0	33.3	42.8	0
Linear i.	0.97	0.70	57.1	100.0	77.8	87.5	0

Table 5.3: Smoothing results for *march*

5.4.3 Evaluation of Smoothing

We shall first illustrate smoothing performance with the single test verb *march*, and then look at the overall performance with the 60 test verbs.

Table 5.3 shows the baseline, add-one, Katz backing-off and linear interpolation smoothing results for *march*. For each method it lists the KL, RC and system results, and the number of correct unseen SCFs, as compared to the gold standard. Table 5.4 shows the ranking of correct (gold standard) SCFs for *march* in the *unfiltered* distributions obtained using the baseline method and the three smoothing methods[6]. The fifth column includes, for comparison, the correct SCF ranking for *march* in the gold standard. The highest ranked SCFs are listed first, the lowest last. SCFs missing in the baseline distribution which occur in the gold standard distribution are indicated using bold font.

As these results illustrate, add-one smoothing preserves the ranking of SCFs which appear in the baseline distribution. It therefore has little or no impact on system performance. With *march*, the KL and RC scores worsen. This is due to the method assigning all missing SCFs a uniform probability. Thus although add-one detects all SCFs unseen in data from the hypothesis generator, it may not improve results for low frequency SCFs.

Katz backing-off preserves the ranking of the most frequent SCFs. As a consequence, results for high frequency SCFs are rarely affected and there is little change in the system performance. With *march*, KL worsens slightly, while RC shows improvement. The reason is apparent in table 5.4. After smoothing with Katz backing-off, the newly-detected SCFs 74, 78 and 76 are correctly ranked higher than SCF 28, which is also newly detected. In addition, SCF 49 appears, correctly, lower in the ranking scale. Unlike add-one smoothing, Katz backing-off can thus correct the ranking of low frequency SCFs, depending on the accuracy of the back-off estimates.

Unlike add-one smoothing and Katz backing-off, linear interpolation also affects high frequency SCFs. With *march*, system results, as well as KL and RC, improve significantly. As illustrated in table 5.4, linear interpolation correctly lowers the high frequency SCF 24 in the ranking list, while raising 87 higher. It also gets the ranking of the lower frequency SCFs 49 and 28 right. Thus when back-off estimates are accurate, one may expect good overall results with linear interpolation.

Table 5.5 gives average results for all the 60 test verbs using each smoothing method.

[6] Note that all incorrect SCFs are omitted in this table, as these do not occur in the gold standard.

Baseline	**Add-one**	**Katz b.**	**Linear i.**	**Correct**
22	22	22	22	87
24	24	24	87	22
87	87	87	24	**76**
95	95	**74**	**74**	**78**
49	49	**78**	**78**	**74**
	27, 76, 78, 74	95	95	95
		76	**76**	24
		27	49	49
		49	**27**	**27**

Table 5.4: Ranking of correct gold standard SCFs for *march* in acquired *unfiltered* distributions. (Note that all incorrect SCFs are omitted in this table).

			System results				Unseen
Method	KL	RC	Rank A. (%)	Precision (%)	Recall (%)	F	SCFs
Baseline	0.63	0.72	79.2	78.5	63.3	70.1	151
Add-one	0.64	0.74	79.0	79.1	64.8	71.2	0
Katz b.	0.61	0.75	79.0	76.4	67.6	71.7	3
Linear i.	0.51	0.82	84.4	87.8	68.7	77.1	3

Table 5.5: Average results with different methods using semantically motivated back-off estimates for smoothing

These results indicate that both add-one smoothing and Katz backing-off improve the baseline performance only slightly. Katz backing-off shows clearer improvement, demonstrating that it is advantageous to use back-off estimates to obtain the likelihood of low and zero frequency SCFs. However, linear interpolation outperforms both these methods, achieving better results on all measures. The improved KL measure indicates that the method improves the overall accuracy of SCF distributions. The results with RC and system accuracy show that it helps to correct the ranking of SCFs. That both precision and recall show clear improvement over baseline results demonstrates that linear interpolation can successfully be combined with the filtering (i.e. thresholding) method employed. These results seem to suggest that a smoothing method which affects both highly ranked SCFs and those of low frequency is profitable for this task.

For comparison, we re-ran these experiments using the unconditional SCF distribution of all verbs as back-off estimates for smoothing. These estimates were obtained by extracting the number of verbs which are members of each SCF in the ANLT dictionary. Average results for the 60 test verbs given in table 5.6 show that, with these estimates, we obtain worse results than with the baseline method. Thus while such estimates provide an easy solution to the sparse data problem, they can actually degrade the accuracy of verbal acquisition. This is in agreement with the well-known view of Gale and Church (1990): poor estimates of context are worse than none.

			System results				Unseen
Method	KL	RC	Rank A. (%)	Precision (%)	Recall (%)	F	SCFs
Baseline	0.63	0.72	79.2	78.5	63.3	70.1	151
Katz b.	0.68	0.69	77.2	75.2	61.7	67.8	0
Linear i.	0.79	0.64	76.7	71.4	64.1	67.6	0

Table 5.6: Average results using the unconditional distribution as back-off estimates for smoothing

	KL			RC		
Verb Class	BL	LI	%	BL	LI	%
9. Verbs of Putting	0.70	0.66	+6	0.68	0.70	+3
11. Verbs of Sending and Carrying 12. Verbs of Exerting Force	0.64	0.50	+22	0.72	0.96	+33
13. Verbs of Change of Possession	0.61	0.60	+2	0.61	0.75	+23
34. Verbs of Assessment	0.81	0.62	+23	0.61	0.70	+15
36. Verbs of Social Interaction	0.65	0.58	+11	0.72	0.80	+11
42. Verbs of Killing	0.69	0.67	+3	0.91	0.95	+4
44. *Destroy* Verbs	0.95	0.20	+79	0.70	0.97	+39
48. Verbs of Appearance, Disappearance and Occurrence	0.14	0.17	-21	0.91	0.83	-9
51. Verbs of Motion	0.66	0.58	+12	0.56	0.66	+18
55. Aspectual Verbs	0.48	0.54	-13	0.86	0.89	+3

Table 5.7: Baseline and linear interpolation results for the verb classes

5.4.4 Evaluation of Back-off Estimates

Table 5.5 shows that, in the above experiment, the semantically motivated back-off estimates helped significantly to reduce the sparse data problem. While a total of 151 gold standard SCFs were unseen in the data from the hypothesis generator, only three were unseen after smoothing with Katz backing-off or linear interpolation. Table 5.7 displays individual results for the different verb classes. It lists the results obtained with KL and RC using the baseline method (BL) and linear interpolation (LI) (with the semantically motivated back-off estimates). It also gives the percentage linear interpolation improved (+) or worsened (-) the baseline KL and RC scores. As linear interpolation is highly sensitive to accuracy of back-off estimates, examining these results allows us to consider the accuracy of the back-off estimates for each verb class.

Out of ten verb classes, eight show improvement with linear interpolation, with both KL and RC. "*Destroy* Verbs" show the biggest improvement over baseline results, while "*Verbs of Killing*" show the smallest improvement. From the 51 individual test verbs included in these eight classes, only two show worse results after smoothing with linear interpolation. The first is the "Putting" verb *place*, which takes noticeably fewer SCFs than the other "Putting" verbs examined. Back-off estimates for this verb class include high-ranking SCFs not taken by *place*. This results in false positives, degrading performance slightly. The second verb is the "Motion" verb *dance*. *Dance* takes SCFs typical to "Verbs of Motion", but the ranking of these SCFs differs from the ranking of those in back-off estimates. One reason for this is that *dance* occurs in the corpus data analysed exceptionally frequently in idiomatic expressions, such as *we danced the night away.*

Two verb classes - "Aspectual Verbs", and "Verbs of Appearance, Disappearance and Occurrence" - show worse results when linear interpolation is used. The problem with "Aspectual Verbs" is that the class contains verbs taking sentential complements. Two verbs examined (*begin* and *start*) occur frequently with sentential complements, while three others (*end, terminate* and *complete*) do not take them at all. According to Levin (1993), these two verb classes need further classification before full semantic account can be given. As Levin does not classify verbs on the basis of their sentential complement-taking properties, further classification is required before we can obtain accurate SCF estimates for this type of verb.

The problem with "Verbs of Appearance, Disappearance and Occurrence" is more specific to the verb class. For example, "Disappearance Verbs" (*disappear* and *vanish*) take noticeably fewer SCFs than "Appearance Verbs" (*arise* and *emerge*). In addition, verbs belonging to the different (and even same) subclasses seem to differ greatly in terms of SCFs they take. For example, from the SCFs taken by *arise* and *emerge*, less than half are shared by both, although the verbs belong to the same subclass. Levin remarks that the definition of this verb class may be too loose, which may explain the poor results.

The poor results with the two verb classes suggest that it is worth examining the degree of SCF correlation between verbs from different subclasses before deciding on the final (sub-)class for which to obtain the estimates. As seen with the eight other verb classes examined, more often than not, back-off estimates can successfully be based on a broad Levin class. As seen with the combined verb class (Levin classes 11 and 12), estimates can also be built using verbs from different Levin classes, provided that the classes are similar enough. Examination of the degree of SCF correlation beforehand would, however, be a useful precaution to guarantee the accuracy of back-off estimates.

Interestingly, for the eight verb classes which show improvement with linear interpolation, the average optimal value of λ_2 used in smoothing was 0.5 (the values for λ_j were obtained by optimisation, see section 5.3.2). Thus when back-off estimates were accurate, best results (on average) were obtained by giving conditional and back-off distributions equal weight. The fact that values higher than 0.5 for λ_2 generally did not further improve (but gradually degraded) performance demonstrates that automatic SCF acquisition is vital and that we could not obtain an accurate lexicon merely by using back-off estimates. Conversely, the fact that the average optimal value for λ_2 was as high as 0.5 demonstrates the utility of semantically motivated back-off estimates in guiding SCF acquisition.

5.5 Discussion

The experimental evaluation reported in the above section shows that the new method of hypothesis selection outperforms the baseline MLE method, addressing the sparse data problem effectively and producing better overall results. Smoothing with linear interpolation, which gives more emphasis on back-off estimates than the other smoothing methods, produces especially good results. This is a direct indication of the accuracy of the verb class specific estimates employed.

The proposed method seems promising but could it be applied to benefit large-scale SCF acquisition? This would require (a) defining the set of semantic verb classes across the entire lexicon, (b) obtaining back-off estimates for each verb class, and (c) implementing a method capable of automatically assigning verbs to semantic classes.

Verbs could be assigned to semantic classes via WordNet, using a method similar to that employed by Dorr (1997). Defining a comprehensive set of verb classes is realistic as well, given that Levin's classification provides a good starting point and that work on refining and extending this classification is already available (e.g. Dang *et al.*, 1998; Dorr, 1997). The manual effort needed to obtain the back-off estimates was quite high for this preliminary experiment, yet, our investigation[7] shows that the total number of semantic classes across the whole lexicon is unlikely to exceed 50. Although some broad Levin classes must be broken down into subclasses, many are similar enough in terms of SCF distributions to be combined. The additional effort required to apply the method to benefit large-scale SCF acquisition thus seems justified, given the accuracy enhancement reported.

5.6 Summary

In this chapter, we proposed a method for constructing verb class specific back-off estimates for SCF acquisition and a new semantically motivated approach for hypothesis selection which combines MLE thresholding and smoothing with the back-off estimates. We reported experiments which demonstrated that the back-off estimates can be used to significantly improve SCF acquisition through the approach employed for hypothesis selection, when linear interpolation is employed for smoothing. Finally, we considered the work required for extending this method to large-scale SCF acquisition. We concluded that, despite the manual effort involved in constructing the back-off estimates, the task seems justified, given the improvements reported.

[7] See section 6.3.1 for the method of investigation.

Chapter 6

Semantically Motivated Subcategorization Acquisition

6.1 Introduction

This chapter concerns semantically motivated lexical acquisition: specifically, the use of a priori knowledge about verb semantics in guiding the process of automatic SCF acquisition. We shall start by looking at some related work on this topic (section 6.2). In section 6.3, we outline our own approach. This involves describing how the novel approach for hypothesis selection introduced in the previous chapter was refined further and integrated as part of automatic large-scale SCF acquisition. Experiments for evaluation of the approach refined are reported in section 6.4. Section 6.5 discusses further work and section 6.6 contains a summary and our conclusions.

6.2 Related Work

Our work on semantically motivated SCF acquisition relates to the (computational) linguistic research (Fillmore, 1968; Grimshaw, 1990; Hale and Keyser, 1993; Jackendoff, 1990; Levin, 1993; Levin and Rappaport Hovav, 1996; Pinker, 1989) which suggests that there is a close relation between the underlying lexical-semantic structures and their associated syntactic behaviour. While lexical-semantic structures may fall short of providing full semantic inference, they can provide a robust basis for the development of language-processing functions and an analysis more useful than the merely syntactic (Dorr, 1997). That some semantic components can be identified with syntactic behaviour opens up the possibility of inferring semantics of a word on the basis of its syntactic behaviour, and the syntax of a word on the basis of its semantic behaviour. This possibility is of special interest for lexical acquisition. In this section, we discuss how information about diathesis alternations and verb semantic classes has so far been used to aid the process of lexical acquisition (Ribas, 1995; Poznanski and Sanfilippo, 1995; Korhonen, 1998). In chapter 7 (section 7.2.2), we will consider further ways of exploiting the syntax-semantics link in subcategorization acquisition.

Ribas (1995) used diathesis alternations to aid acquisition of selectional preferences. He did this by combining the argument head data which occur in the argument slots of the alternating variants involved in a diathesis alternation, and by acquiring the selectional preferences from the combined data. Ribas experimented with one alternation only, the passive alternation with the verb *present*:

(29) a *She presents great risks*
b *The challenge being presented to us by Tim*

Selectional restrictions for the subject and object slots of *present* were acquired from the wsj corpus data. Three different methods were then applied to assess the benefit of alternation information. Method 1 involved acquiring selectional restrictions specific to different argument slots, regardless of the passive alternation. Method 2 involved detecting the passive alternation and acquiring selectional restrictions specific to argument slots with the same semantic role. Method 3 involved detecting the passive alternation, combining argument head data from the alternating slots, and acquiring selectional restrictions specific to the combined data. The latter method would, for example, combine *she* in (29a) and *Tim* in (29b) before acquiring selectional restrictions.

Ribas evaluated the three methods on a wsd task. Method 3 achieved the best results, outperforming the others both in precision and recall. While this is an encouraging result, Ribas mentions two problems that would need to be tackled if the method were to be extended beyond the passive alternation: the low frequency of diathesis alternation patterns in the wsj data, and the difficulty of detecting alternation patterns on purely syntactic grounds.

Poznanski and Sanfilippo (1995) used semantic class information to aid wsd. They presented a system capable of individuating dependencies between the verb semantic classes and their associated scfs. The system extracts scf tokens from the Penn Treebank, supplements the scf tokens with semantic tags from the lloce thesaurus (McArthur, 1981), and converts the scf tokens into scf types. A scf type consists of a verb stem associated with one or more lloce semantic tags. Semantic ambiguity arising from multiple tag assignments is removed by using the lloce collocational information. The codes of word stems, which according to the collocational information are incompatible with the type of scf in which they occur, are filtered out. (30a) shows the scf token for *deny aliens state benefits*. The scf type for this token is shown in (30b), where *deny* is associated with two potential semantic tags: "c193"-refuse and "g127"-reject. The disambiguator chooses the latter (30c) as, according to the lloce collocational information, *deny* can only take ditransitive scf in the *refuse* sense.

```
(30)  a SCF token: ((DENY)
                     (NP (ALIENS NNS))
                     (NP (*COMPOUND NOUN* (STATE NN) (BENEFITS NNS))))

      b SCF type:  (("deny" ("C193"-refuse "G127"-reject))
                     ((*NP* ("C"-people_and_family))
                      (*NP* ("N"-general_and_abstract_terms"))))

      c Disambiguated SCF type: (("deny" ("C193"))
                                  ((*NP* ("C"))
                                   (*NP* ("N"))))
```

This approach was evaluated on a set of 1335 SCF tokens which were converted into 817 SCF types. The system managed to reduce ambiguity in over half of the SCF types and totally disambiguate over 16%, providing unique correspondence between a semantic class and a SCF in 346 cases. This demonstrates the utility of semantic class information in aiding lexical disambiguation. In addition, preliminary results were reported which showed that verbs associated with similar semantic codes took similar SCFs. For instance, the verbs associated with *putting* and *taking*, and *pulling* and *pushing* senses showed a higher than average tendency for SCF NP-PP. Poznanski and Sanfilippo discussed the possibility of using semantic class information to predict unseen SCF options, but reported no work on this.

Korhonen (1998) used alternations to improve automatic SCF acquisition. The approach involved correcting the performance of the statistical filter of Briscoe and Carroll's SCF acquisition system[1] by addition of information about likely diathesis alternations. The basic idea was to make use of likely correlations between pairs of SCFs. For example, an English verb which takes a NP-S complement (*It bothered John that Bill was so clever*) is unlikely also to take a S complement (**It bothered that Bill was so clever*). If a hypothesis generator proposes these two SCFs for the same verb, one is likely to be wrong and should be dropped during hypothesis selection.

Korhonen examined the errors with SCFs in system output and arrived at nine alternations which could aid correction of these errors. In addition, a large set of alternations was constructed automatically by considering correlations between all possible SCF types in the ANLT dictionary. The alternations were expressed as directional rules of the form:

(31) SCF A ⟶ SCF B

Each rule was assigned a probability by calculating the number of ANLT verb types in both SCF A and SCF B, and dividing this by the number of verb types in SCF A.

The alternation rules were applied at the filtering phase of the SCF acquisition process. The systems hypothesis generator was run as usual, and the BHT filter was used to build SCF entries and assign each entry a probability[2]. However, no confidence

[1] See section 2.5.3 for the system description and details of the statistical filter.

[2] BHT assigns each verb and SCF combination $P(m+, n, p^e)$, which is the probability that m or more occurrences of cues for scf_i will occur with a verb which is not a member of scf_i, given n occurrences of that verb. See section 2.5.3 for details of this calculation.

threshold was set on the probabilities, but instead, the alternation probabilities were applied. This was done according to the following principle:

Given an alternation rule SCF A $\longrightarrow$ SCF B, if both SCF A and SCF B are hypothesised for a verb, the probability assigned to SCF A by BHT is improved by the probability of the alternation rule. However, if SCF A is hypothesised for a verb but SCF B is not, the probability assigned to SCF A by BHT is lowered by the probability of the alternation rule.

Let $p_{scf_{\mathrm{A}}}$ be the probability of SCF A given a verb according to BHT and $p(A \longrightarrow B)$ the probability of an alternation rule. If SCF B is also hypothesised for the verb, the revised probability of SCF A is

$$P_{scf_{\mathrm{A}}} = p_{scf_{\mathrm{A}}} - w((p_{scf_{\mathrm{A}}}) \cdot p(A \longrightarrow B)) \tag{6.1}$$

If SCF B is not hypothesised for the verb, the revised probability of SCF A is

$$P_{scf_{\mathrm{A}}} = p_{scf_{\mathrm{A}}} + w((p_{scf_{\mathrm{A}}}) \cdot p(A \longrightarrow B)) \tag{6.2}$$

where w is defined empirically. After revising the probabilities assigned by BHT in the above way, entries are filtered using a confidence threshold of 0.05.

Suppose then we have the alternation rule[3] SCF 49 $\longrightarrow$ SCF 24 with probability .38. If BHT assigns SCF 49 the probability of 0.08, this SCF would normally be rejected, as $0.08 \geq 0.05$. Using the method described above, however, the SCF would be accepted if SCF 24 were also hypothesised for the verb:

$$0.0496 = 0.08 - 1(0.08 \cdot 0.38)$$

Korhonen evaluated this approach with 23 unseen test verbs, using the same evaluation method and corpus data as Briscoe and Carroll (1997), described in section 2.5.2. The large set of automatically derived alternation rules from ANLT improved the system's ranking accuracy and type precision by 4%, and type recall by 5% over the baseline results obtained with the original system.

As our review indicates, little work exists on using verb semantic information for guiding automatic lexical acquisition. Our approach most closely resembles that of Korhonen (1998) in that we also use semantic knowledge to aid SCF acquisition, and do this at the hypothesis selection phase of the process. Our approach here is, however, more knowledge-driven. Instead of using empirical information about likely alternations, we classify verbs into semantic classes and use probabilistic information related to these classes. We also employ a more accurate method for hypothesis selection which uses verb class specific back-off estimates, enabling us to exploit semantic class information to detect unseen SCFs. The idea of detecting SCFs on the basis of semantic information was earlier raised by Sanfilippo (1994) and Poznanski and Sanfilippo (1995), but to our knowledge, it has not yet been applied to automatic SCF acquisition.

[3] According to the SCF classification used, SCF 49 is equivalent to NP-PP frame and 24 to NP frame.

6.3 Method for Semantically Motivated SCF Acquisition

In chapter 5, we proposed a new approach for semantically motivated hypothesis selection. In this section, we describe how the method was further refined and extended to suit large-scale automatic SCF acquisition.

The basic idea of the method, as outlined in chapter 5, is to identify the semantic class of a verb, use the SCF acquisition system to hypothesise a conditional SCF distribution for the verb, smooth this distribution with the back-off estimates of the respective semantic class, and use a simple technique for filtering SCFs which applies a threshold to the resulting set of probability estimates. This method requires (a) semantic verb classes, (b) verb class specific back-off estimates, (c) a technique for identifying semantic classes of verbs, and (d) a filtering method which employs the back-off estimates. In chapter 5, we proposed methods for (a), (b) and (d). In this chapter, we shall propose modifications to the methods for (a) and (b), but adopt the method for (d) as it stands. In other words, we shall employ the filtering approach introduced in section 5.3.2 as it stands, and use it primarily with the smoothing technique that proved best (i.e. linear interpolation). In chapter 5, no method was proposed for (c). Rather, verbal participation in semantic classes was identified manually. In this section, we propose a technique which does this automatically.

The following sections describe these changes and extensions made to the basic method. Section 6.3.1 describes (a) our approach with semantic classes and section 6.3.2 gives details of (b) the technique used for obtaining back-off estimates for the classes. Section 6.3.3 introduces (c) the method capable of automatically assigning verbs to semantic classes. Finally, section 6.3.4 describes the application of methods reported in the three previous sections. It reports the work completed on semantic classes, back-off estimates and semantic classification of verbs.

6.3.1 Semantic Classes

In section 5.2, we proposed basing our semantic classes on Levin classes. The latter provided us with a good starting point for large-scale SCF acquisition as well. Although not comprehensive in breadth or depth of coverage, the classes cover a substantial number of diathesis alternations occurring in English. In addition, work on refining and extending this classification is under way (Dang *et al.*, 1998; Dorr, 1997; Korhonen, Appendix C).

Dang *et al.* (1998) have refined some Levin classes into *intersective* classes. To remove the overlap between the extant classes, they have created intersective classes for those Levin verbs which share membership of more than one Levin class. For example, an intersective class was formed for the verbs *pull, tug, show* and *push* that are triple-listed in the Levin classes of "*Split*", "*Push/Pull*" and "*Carry Verbs*". These verbs show characteristic syntactic and semantic behaviour not typical of their original verb classes. Although Dang *et al.* report only preliminary work on a few verb classes, it seems promising: the intersective classes provide a finer-grained classification with more coherent sets of SCFs and associated semantic components.

Dorr (1997) has created new semantic classes for verbs whose syntactic behaviour differs from the syntactic description of existing Levin classes. The creation of new classes is a by-product of her verb classification algorithm which assigns unknown verbs into Levin classes. We shall discuss this algorithm in detail in section 6.3.3. Essentially, the syntactic description of Levin classes, which corresponds roughly to the alternation pairs from Levin (1993), is represented by sets of codes adopted from the LDOCE dictionary. If no Levin class is found to match the LDOCE syntactic description of the unknown verb, a new semantic class is created, and each verb matching its syntactic description is classified as a member. Using this method, Dorr has arrived at 26 novel classes. The majority of these classes concern verb types not covered by Levin, e.g. those taking sentential complements.

We have, in addition, proposed new diathesis alternations not included in Levin (1993), particularly those involving sentential complements. We did this work while collaborating with Diana McCarthy on automatic diathesis alternation detection (see section 7.2.3 for details of this work). The new alternations are discussed briefly in Appendix C of this thesis. They were obtained by manually examining the classification of 163 SCFs employed by Briscoe and Carroll's system and considering possible alternations between pairs of SCFs in this classification. Novel alternations could be used further to refine Levin verb classes and to create new classes for verb types not covered by Levin.

The above work demonstrates that extending Levin's classification to obtain a comprehensive set of verb classes across the entire lexicon is a realistic goal. In the work reported in this chapter, however, we restrict ourselves to employing existing Levin classes.

In chapter 5, we took the broad Levin classes as a starting point. Assuming a broad class whenever possible makes sense, as it minimises the manual effort required in constructing the back-off estimates for each class: obtaining back-off estimates for a broad class is less laborious than obtaining separate estimates for each of its subclasses. The experiments reported in section 5.4.4 showed, however, that while many of the broad classes are distinctive enough in terms of subcategorization, and while some can successfully be combined, others need to be broken down into subclasses. This suggests that we should examine the distinctiveness of Levin classes in terms of subcategorization prior to deciding on the grouping of these classes. We did this in two steps, by examining the

- syntactic similarity between Levin classes (**Step 1**)
- subcategorization similarity between verbs in Levin classes (**Step 2**)

Step 1 gives us an indication of whether the verb senses involved in the classes are syntactically similar enough. It also helps to identify the Levin classes which need further refinement. Step 2 complements Step 1, as the syntactic information included in Levin (1993) is not always conclusive and does not provide any information about the relative frequency of SCF options. In addition, it allows us to examine the degree of SCF correlation between the verb form specific SCF distributions we are actually concerned with. The subsequent two sections describe how we proceeded with Steps 1 and 2, respectively.

Class	Syntactic Pattern	LDOCE Codes
30.1	0-[np,v]	I
***See* Verbs**	0-[np,v,np,pp(for)]	T1-FOR D1-FOR
	0-[np,v,pp(at)]	I-AT I3-AT L9-AT WV4-AT
	1-[np,v,np]	T1 L1
	1-[np,v,np,vp]	V4 V4-FROM V4-WITH X4 V2
	1-[np,v,np,pp(in)]	D1-IN T1-IN
	1-[np,v,s comp]	T5 I5 X5
	1-[np,v,vp]	I4 T4 L4 T2 I2 WV2
30.2	0-[np,v]	I
Sight Verbs	0-[np,v,s comp]	T5 I5 X5
	0-[np,v,pp(at)]	I-AT I3-AT L9-AT WV4-AT
	1-[np,v,np]	T1 L1
	1-[np,v,np,vp]	V4 V4-FROM V4-WITH X4 V2
30.3	0-[np,v,np]	T1 L1
Peer Verbs	0-[np,v,s comp]	T5 I5 X5
	1-[np,v,pp(around)]	L9 I
	1-[np,v,pp(at)]	I-AT I3-AT L9-AT WV4-AT
	1-[np,v,pp(into)]	I-INTO
	1-[np,v,pp(on)]	I-ON L9-ON I-UPON
	1-[np,v,pp(through)]	I-THROUGH
	1-[np,v,pp(to)]	I-TO
30.4	0-[np,v]	I
Stimulus Subject	1-[np,v,adjective]	L7 WA4
Perception Verbs	1-[np,v,adjective,pp(to)]	L7

Table 6.1: LDOCE codes for "Verbs of Perception"

Step 1: Syntactic Similarity between Levin Classes

For Step 1, we employed Dorr's source of LDOCE codes for Levin classes[4]. Dorr (1997) extracted automatically basic syntactic patterns from all the sentences in Levin's book. The patterns were mapped onto LDOCE codes and grouped into canonical and prohibited codes for each class. We used Dorr's LDOCE codes to determine the syntactic similarity between Levin classes. This was done by considering the degree of intersection between the codes for the classes.

For example, table 6.1 shows the LDOCE codes for each of the four subclasses of the broad Levin class of "Verbs of Perception". The first column of the table indicates the Levin subclass in question, the second lists a syntactic pattern extracted from a Levin sentence and the third gives the LDOCE code corresponding to the pattern. Canonical and prohibited LDOCE codes are prefixed as "1-" and "0-", respectively. By examining the codes, we can tell that syntactic descriptions of subclasses differ significantly. Firstly, no LDOCE code (canonical or prohibited) is shared by all four subclasses. The only intersection with canonical codes occurs between subclasses 30.1 and 30.2, which share two canonical codes (T1 L1 and V4 V4-FROM V4-WITH X4 V2). Classes 30.1, 30.2 and 30.4 share one prohibited code (I), classes 30.1 and 30.2 share one (I-AT I3-AT L9-AT WV4-AT) and classes 30.2 and 30.3 one (T5 I5 X5). However, a

[4] We are indebted to Bonnie Dorr for the use of these codes. We adopted the codes as they stand but removed duplicate and uncertain code assignments. See Procter (1978) for a detailed description of LDOCE grammatical codes and Dorr (1997) for further information.

Class	**Syntactic Pattern**	**LDOCE Codes**
11.1 ***Send* Verbs**	0-[np,v,pp(at),pp(to)]	I-TO I-AT I3-AT L9-AT WV4-AT
	0-[np,v,pp(to)]	I-TO
	1-[np,v,np]	T1 L1
	1-[np,v,np,np]	D1 X1
	1-[np,v,np,pp(from)]	D1-FROM T1-FROM
	1-[np,v,np,pp(to)]	D1-TO T1-TO WV5-TO
	1-[np,v,np,pp(with)]	D1-WITH X7-WITH T1-WITH WV5-WITH X9-WITH
	1-[np,v,pp(from),pp(to)]	I-FROM I-TO
11.2 ***Slide* Verbs**	0-[np,v,np,pp(with)]	D1-WITH X7-WITH T1-WITH WV5-WITH X9-WITH
	0-[np,v,pp(at),pp(to)]	I-TO I-AT I3-AT L9-AT WV4-AT
	1-[np,v,np]	T1 L1
	1-[np,v,np,np]	D1 X1
	1-[np,v,np,pp(across)]	T1-ACROSS
	1-[np,v,np,pp(to)]	D1-TO T1-TO WV5-TO
	1-[np,v,pp(across)]	L9
	1-[np,v,pp(at)]	I-AT I3-AT L9-AT WV4-AT
	1-[np,v,pp(from),pp(to)]	I-FROM I-TO
	1-[np,v,np,pp([away,from])]	X9
11.3 ***Bring* and *Take***	0-[np,v,np,adjective]	X7
	0-[np,v,pp(at),pp(to)]	I-TO I-AT I3-AT L9-AT WV4-AT
	0-[np,v,pp(to)]	I-TO
	1-[np,v,np,np]	D1 X1
	1-[np,v,np,pp(from)]	D1-FROM T1-FROM
	1-[np,v,np,pp(to)]	D1-TO T1-TO WV5-TO
	1-[np,v,np,pp(with)]	D1-WITH X7-WITH T1-WITH WV5-WITH X9-WITH
11.4 ***Carry* Verbs**	0-[np,v]	I
	0-[np,v,pp(at),pp(to)]	I-TO I-AT I3-AT L9-AT WV4-AT
	0-[np,v,pp(to)]	I-TO
	1-[np,v,np]	T1 L1
	1-[np,v,np,pp(from)]	D1-FROM T1-FROM
	1-[np,v,np,pp(to)]	D1-TO T1-TO WV5-TO
	1-[np,v,np,pp(with)]	D1-WITH X7-WITH T1-WITH WV5-WITH X9-WITH
	1-[np,v,pp(against)]	I-AGAINST
	1-[np,v,pp(at)]	I-AT I3-AT L9-AT WV4-AT
	1-[np,v,pp(from),pp(to)]	I-FROM I-TO
	1-[np,v,pp(to),pp(with)]	I-TO I-WITH I3-WITH L9-WITH
11.5 ***Drive* Verbs**	0-[np,v]	I
	0-[np,v,np,pp(with)]	D1-WITH X7-WITH T1-WITH WV5-WITH X9-WITH
	0-[np,v,pp(at)]	I-AT I3-AT L9-AT WV4-AT
	0-[np,v,pp(at),pp(to)]	I-TO I-AT I3-AT L9-AT WV4-AT
	0-[np,v,pp(to)]	I-TO
	1-[np,v,np]	T1 L1
	1-[np,v,np,pp(from)]	D1-FROM T1-FROM
	1-[np,v,np,pp(to)]	D1-TO T1-TO WV5-TO
	1-[np,v,pp(from),pp(to)]	I-FROM I-TO

Table 6.2: LDOCE codes for “Verbs of Sending and Carrying”

Class	Syntactic Pattern	LDOCE Codes
12. **Verbs of** **Exerting Force**	0-[np,v]	I
	1-[np,v,np]	T1 L1
	1-[np,v,np,adjective]	X7
	1-[np,v,np,pp(against)]	T1-AGAINST
	1-[np,v,np,pp(through)]	X9 T1-THROUGH
	1-[np,v,np,pp([away,from])]	X9
	1-[np,v,pp(against)]	I-AGAINST
	1-[np,v,pp(at)]	I-AT I3-AT L9-AT WV4-AT
	1-[np,v,pp(on)]	I-ON L9-ON I-UPON
	1-[np,v,pp(through)]	I-THROUGH
	1-[n]	N

Table 6.3: LDOCE codes for "Verbs of Exerting Force"

canonical code for one class shows up as a prohibited code for another class, and vice versa. Secondly, the number of codes taken by the different subclasses varies greatly. For example, class 30.3 takes six canonical codes, while class 30.2 takes only two. These observations suggest that the syntactic descriptions of the subclasses are so dissimilar as to merit our obtaining the back-off estimates specific to the subclasses, rather than to the broad class of "Perception" verbs.

Table 6.2 lists the LDOCE codes for each of the four subclasses of Levin's "Verbs of Sending and Carrying". With this broad class, the syntactic descriptions of subclasses prove more similar. All five subclasses share one canonical code (D1-TO T1-TO WV5-TO). In addition, four subclasses share three canonical codes (I-FROM I-TO, D1-FROM T1-FROM and T1 L1) and three share one (D1 X1). None of the latter are found among the prohibited codes of the other classes. With prohibited codes, one code is shared by all five subclasses (I-TO I-AT I3-AT L9-AT WV4-AT) and another by four subclasses (I-TO). Although some prohibited codes occur as canonical codes for other classes, the intersection of both canonical and prohibited codes is fairly extensive. This suggests that the broad class of "Sending and Carrying Verbs" is syntactically coherent enough to provide an adequate basis for back-off estimates.

The above examples illustrate typical choices between more or less specific Levin classes. In addition, some semantically similar broad Levin classes are syntactically similar enough to be combined. For example, "Verbs of Sending and Carrying" and "Verbs of Exerting Force" are semantically fairly similar: some Levin verbs are cross-listed in these two classes, as they share the semantic component of exertion of force. To find out whether these classes could be combined, we compare their syntactic descriptions. The LDOCE codes for "Verbs of Sending and Carrying" were shown in table 6.2, and those for "Verbs of Exerting Force" are listed in table 6.3. The two classes share one prohibited and four canonical codes. Only one canonical code for class 12 is found among prohibited codes for class 11 (with one subclass only). The two broad classes seem thus syntactically similar enough to be combined.

Step 2: Subcategorization Similarity between Verbs in Levin Classes

For Step 2, we chose representative verbs from Levin's classification. These were chosen at random, subject to the constraint that they occurred frequently enough in corpus data, represented different subclasses of a broad Levin class (when applicable), and that their most frequent sense in WordNet involved the Levin class in question. The SCF distributions for these verbs were obtained by manually analysing c. 300 occurrences of each verb in the BNC data, using the SCF classification in Appendix A. After this, the resulting SCF distributions were compared in terms of

- the intersection of shared SCFs
- the dissimilarity of distributions
- the similarity in ranking of SCFs in distributions

Table 6.4 shows the SCFs as code numbers for "Sending and Carrying" verbs *send, ship, bring* and *carry* and those for "Exerting Force" verbs *push* and *pull*, as obtained from manual analysis. The different "Sending and Carrying" verbs take a total of 21 different SCFs, 5 of which are shared by all four verbs, a further 3 by three verbs and 5 by two verbs. The average overall KL distance between the different distributions is 0.6 and the average RC, 0.56. The latter results are better than those obtained when correlating the distributions against the unconditional distribution of all verbs in English. These observations support those made by examining verb class similarity: the Levin class of "Verbs of Sending and Carrying" seem distinctive enough in terms of subcategorization.

To examine whether this class could be combined with the other broad class, "Verbs of Exerting Force", the merged SCF distribution of the four "Sending and Carrying" verbs is compared with the merged SCF distribution of the two "Force Exerting" verbs. From a total of 23 different SCFs occurring in the two distributions, 21 occur in both. The average KL distance between the distributions is 0.47 and the average RC, 0.51. These figures again support the observations made earlier with Step 1: the two Levin classes are syntactically similar enough to be combined.

6.3.2 Constructing Back-off Estimates

We adopted the method proposed for constructing back-off estimates in section 5.2 as it stands, with one exception. For some semantic classes, not enough suitable Levin verbs were found that would occur frequently enough in corpus data. In these cases, instead of using the SCF distributions of the ideal 4-5 verbs for constructing the back-off estimates, we used as many as possible.

6.3.3 Assigning Verbs to Semantic Classes

In the work reported so far, verbs were manually assigned to semantic classes. We shall now describe a method we used for automatic classification of verbs. This involves assigning verbs to semantic classes via WordNet. Although WordNet's semantic

send	***ship***	***bring***	***carry***	***pull***	***push***
49	49	76	76	76	76
56	77	56	24	49	49
37	24	24	49	24	24
76	122	120	77	78	77
77	87	27	78	77	22
24	78	31	27	87	87
53	22	49	87	22	74
27	76	122	122	74	78
122	37	74	30		27
150	56	77	74		25
87	3	69	22		53
35	95				3
29					122
					112

Table 6.4: SCFs for "Verbs of Sending and Carrying" and "Exerting Force"

organization does not always go hand in hand with syntactic information, Dorr and Jones (1996a, 1996b) and Dorr (1997) have demonstrated that synonymous verbs in WordNet exhibit syntactic behaviour similar to that characterised in the classification system of Levin. This enables association of verbs with semantic classes on the basis of their WordNet synonyms. Our semantic verb classification approach resembles that previously taken by Dorr (1997). We shall begin by reviewing this related work, after which we introduce our own method.

Previous Work

Dorr's (1997) verb classification algorithm is a refined version of those proposed in Dorr and Jones (1996a, 1996b). It assigns each unknown verb to a semantic class by examining the verb's synonyms from WordNet and selecting those whose Levin class is associated with syntactic information matching that of the unknown verb. The syntactic information is expressed as LDOCE codes[5]. The classification algorithm works as follows:

Step 1: If a given verb V is in Levin's index, it is classified directly.

Step 2: Otherwise, V's WordNet synonyms are extracted.

Step 3: If none of V's WordNet synonyms is in Levin's index, V is set aside for later application of the algorithm (after one or more of its synonyms is classified).

Step 4: A candidate set of semantic classes (from Levin's index) corresponding to the synonyms in V's synset(s) is produced.

Step 5: If V's LDOCE codes do not match the canonical LDOCE codes for any semantic class associated with the WordNet synonyms, a new class is created.

[5] See section 6.3.1 for description of Dorr's source of LDOCE codes for Levin classes.

Step 6: If *V*'s LDOCE codes match the canonical LDOCE codes for a semantic class associated with the WordNet synonyms, *V* is included in that class.

The notion of "match" in this algorithm is based on the degree of intersection between *V*'s LDOCE codes and the canonical LDOCE codes for a candidate class. A preference is given to those classes whose prohibited LDOCE codes are not among *V*'s LDOCE codes. A preference is also given to the classes containing the highest number of matching WordNet synonyms. The algorithm is run iteratively: after 100-200 verbs are classified, the procedure is re-run on the remaining set of unknown verbs with the larger database of semantic classes.

As an example, let us consider the semantic classification of *swear* according to Dorr's algorithm. The LDOCE specification of this verb is I I-AT T1 T1-ON T1-TO T3 T5. Step 4 of the classification algorithm extracts candidate Levin classes associated with the WordNet synonyms of this word: (1) class 29.4 "*Declare* Verbs", (2) class 29.5 "Conjecture Verbs", (3) class 37.7 "*Say* Verbs", and (4) class 48.1.2 "Reflexive Verbs of Appearance". The canonical LDOCE codes for each of these classes, respectively, are: (1) D1 X1 D3 V3 T6 GO_BE X7-TO_BE X9-TO_BE V3-TO_BE T5 I5 X5, (2) D3 V3 T6 X1-TO_BE X7-TO_BE X9-TO_BE V3-TO_BE T5 I5 X5, (3) D1-TO T1-TO WV5-TO D5-TO T5-TO T5 I5 X5, and (4) D1-TO T1-TO WV5-TO T1 L1. The largest intersection with the canonical LDOCE codes occurs with class 37.7 (T5 T1-TO). Thus step 6 of the algorithm selects 37.7 as the semantic class for *swear.*

Dorr evaluated this approach using a set of 95 verbs not in Levin (1993), taken from the LDOCE control vocabulary (i.e. primitive words used for dictionary entry definition). A total of 135 semantic class assignments were made with the algorithm, with several verbs receiving more than one class assignment. Of these, 61% were hand-verified to be correct. 22% of incorrect assignments were due to syntactic omissions in LDOCE and Levin (1993). In such cases, the relevant WordNet synonym was available, but the canonical/prohibited codes associated with the synonym's class(es) were not specific enough for the class(es) to be selected. The majority of these omissions were caused by missing syntactic codes in LDOCE. Others arose when a relevant syntactic pattern was missing in Levin's data, or when a WordNet synonym was found in Levin's index but in a class irrelevant to the verb under consideration. The remaining 17% of incorrect assignments corresponded to cases where there is a semantic mismatch between WordNet and Levin (1993). In such cases, the WordNet synonyms for an unknown verb corresponded to word senses that are not available in Levin's classification.

Our aim is similar to Dorr's; we also aim to assign verbs to Levin classes via WordNet. Adopting Dorr's approach as it stands would be problematic, however. The first problem has to do with accuracy. As our method is highly dependent on accurate class assignments, the 61% accuracy of assigning verbs to correct classes is not adequate. This problem is coupled with the fact that the approach allows for multiple class assignment. Given the nature of our task, we assign each verb to only one semantic class and, to achieve overall improvement, this needs to be the class related to the verb's most frequent sense. Dorr's algorithm returns no information about which of the assigned classes, if any, corresponds to the verb's most frequent sense. Accordingly, we shall adopt a different approach for semantic classification of verbs. This approach will, however, use some of the techniques employed in Dorr (1997).

Our Approach

While Dorr's (1997) method assigns verbs to semantic classes on the basis of their WordNet synonyms, ours assigns entire WordNet synsets to semantic classes. In our approach, individual verbs receive the semantic class assignment of their synsets. Our objective is to build a more static source where WordNet synsets are associated with different Levin classes. Although static, the source will allow for updating and adding new verbs to WordNet. Verbs added to the existing synsets are classified directly via their synset and the source can be updated to cover novel synsets. Rather than proposing a Dorr-style fully automatic verb classification algorithm which relies solely on MRDs and other lexical resources, we propose a semi-automatic approach which partly draws on such resources. Since the accuracy of class assignments is highly important for us, some allowance for manual intervention is necessary.

Our method comprises two phases which we introduce in the following paragraphs: annotating Levin classification (**Phase I**) and assigning WordNet synsets to semantic classes (**Phase II**) .

Phase I ***Annotating Levin's Classification*** To employ Levin's verb index usefully and to proceed with the present classification task, we need to know which Levin verbs are already classified according to their predominant sense. As a preliminary step, we annotated the index for the first sense verbs. This was done by manually examining each Levin verb, extracting its predominant sense from WordNet, and comparing it with that/those involved with the semantic class(es) of the verb in Levin's classification. For example, Levin lists the verb *convey* with both "*Send* Verbs" and "*Say* Verbs". According to WordNet, the most frequent sense of *convey* is

```
convey, impart -- (make known; pass on, of information)
     => communicate, intercommunicate -- (transmit thoughts or feelings)
      => interact -- (act together or towards others or with others)
       => act, move -- (perform an action)
```

The hypernym nodes of this sense include those of {*communicate, intercommunicate*} and {*interact*}. The sense in question clearly corresponds to the meaning involved with "*Say* Verbs" rather than to that involved with "*Send* Verbs". Thus we dropped the verb from the latter class and preserved it in the former.

Levin associates most, but not all, verbs with a class that corresponds to their first sense. For example, Levin lists *shift* only with "*Send* Verbs", while its predominant sense corresponds rather to that involved with "Verbs of Change of State":

```
switch, change over, shift, turn around -- (make a shift in or exchange of)
        => change, alter
```

We dropped these cases from the index and set them aside to be classified using the method introduced in the next paragraph.

11. Verbs of Sending and Carrying
11.1 *Send* Verbs *airmail*, (drop: *convey*), (drop: *deliver*), *dispatch*, (drop: *express*), *FedEx*, *forward*, *hand*, *mail*, (drop: *pass*), *port*, (drop: *return*), *send*, (drop: *shift*), *ship*, *shunt*, (drop: *slip*), *smuggle*, (drop: *sneak*), *transfer*, *transport*, *UPS*
11.2 *Slide* Verbs (drop: *bounce*, *float*, *move*, *roll*)
11.3 *Bring* and *Take* *bring*, *take*
11.4 *Carry* Verbs *carry*, (drop: *drag*, *haul*), (drop: *heave*, *heft*, *hoist*, *kick*), *lug*, (drop: *push*, *pull*, *schleg*, *shove*, *tow*)
11.5 *Drive* Verbs *barge*, *bus*, *cart*, (drop: *drive*), *ferry*, (drop: *fly*, *row*, *shuttle*), *truck*, (drop: *wheel*, *wire*)
12. Verbs of Exerting Force
draw, (drop: *heave*), *jerk*, *press*, *pull*, *push*, *shove*, *thrust*, *tung*, *yank*

Table 6.5: Annotated Levin classification

For some other verbs, no sense even exists in Levin (1993) which would correspond to the predominant. The majority of these cases concern verb types not properly covered by Levin, such as verbs taking sentential complements. These were dropped from the index as well and set aside for later examination.

The annotated classification for "Sending and Carrying" and "Force Exerting" verbs is shown in table 6.5. The table shows all Levin verbs, those whose predominant sense is involved with the verb classes listed, and those whose predominant sense is not, and which are thus dropped from the classification. The latter are marked as (drop: *verb*).

Phase II ***Assigning WordNet Synsets to Semantic Classes*** WordNet 1.6 includes 10,319 verb forms whose 22,066 senses spread over 12,127 synsets. These latter divide into 15 subhierarchies which represent different semantic domains. The WordNet files which include the verbs for each subhierarchy are listed in table 6.6. Dorr (1997) notes that many of the top level synsets in the hierarchies intersect directly with the Levin classes. For example, "Sending and Carrying" and "Force Exerting" verbs are all found under the same top level synset {*move, displace*}. Furthermore, verbs belonging to the same Levin classes often occur in the synsets of the same subhierarchy. For example, the most frequent senses of the Levin verbs of "Sending and Carrying" and "Force Exerting" are all found in the verb file "38-verb.motion".

Due to this overlap between WordNet and Levin classes, we associated synsets with Levin classes subhierarchy by subhierarchy, starting from the top level synsets, and going further down in the taxonomy when required. The basic idea was to assign each synset to a semantic class by first assigning the majority of its member verbs to a

Verb Files	**Contains Verbs of**
29-verb.body	grooming, dressing and bodily care
30-verb.change	size, temperature change, intensifying, etc.
31-verb.cognition	thinking, judging, analyzing, doubting
32-verb.communication	telling, asking, ordering, singing
33-verb.competition	fighting, athletic activities
34-verb.consumption	eating and drinking
35-verb.contact	touching, hitting, tying, digging
36-verb.creation	sewing, baking, painting, performing
37-verb.emotion	feeling
38-verb.motion	walking, flying, swimming
39-verb.perception	seeing, hearing, feeling
40-verb.possession	buying, selling, owning
41-verb.social	political and social activities and events
42-verb.stative	being, having, spatial relations
43-verb.weather	raining, snowing, thawing, thundering

Table 6.6: WordNet verb files

semantic class, and then choosing the Levin class supported by the highest number of verbs. 'Member verbs' refer here to those which are members of the synset in question and of its hyponym synsets. Thus if a classified synset has hyponym synsets, the latter are classified according to their classified hypernym synset. Our classification algorithm considers only those verbs whose most frequent sense belongs to the synset in question. The algorithm proceeds as follows:

Step 1: If the majority of member verbs of a given synset S are Levin verbs[6] from the same class, classify S directly. (*See Example 1 below*).

Step 2: Otherwise, classify more member verbs (according to Step 4a-d) until the majority are classified, and then go back to Step 1.

Step 3: If the classified verbs point to different Levin classes, examine whether S consists of hyponym synsets (*See Example 2 below*):

(a) If not, assign S to the Levin class supported by the highest number of classified verbs.

(b) If yes, go one level down in the hierarchy and classify the hyponym synsets separately, starting again from Step 1.

Step 4: If S includes no Levin verbs, proceed as follows to classify the majority of member verbs of S (*See Examples 3 and 4 below*):

(a) Extract the predominant sense of a given verb V from WordNet

(b) Extract the syntactic codes from LDOCE relevant to this sense

[6] For the remainder of this chapter, 'Levin verbs' refer to the first sense verbs in the annotated Levin classification.

(c) Examine whether V could be assigned to a Levin class already associated with the other verbs in the

1. same synset
2. possible hypernym synset
3. possible sister synsets

by comparing the LDOCE codes of the sense and Dorr's LDOCE codes of the respective Levin class(es). Given the hypothesised classes, make the final class assignment manually.

(d) If no suitable class is found, re-examine the case after more verbs have been analysed. If the classification remains unsolved, set V aside for later examination, when it might be grouped with other unclassified verbs and assigned to a verb class not covered by Levin[7].

The above algorithm is for the most part automatic, however, identification of LDOCE codes relevant to the sense in question (Step 4b), and the final class assignment (part of Step 4c) are done manually to ensure accuracy of classification.

The following examples illustrate the use of this algorithm to assign hyponym synsets of the top level synset {*move, displace*} to Levin classes[8]:

Example 1: Synset 01328437 has five first sense member verbs, three of which are Levin verbs from the same verb class. The synset is assigned directly to the Levin class of "Verbs of Sending and Carrying".

```
ship => Verbs of Sending and Carrying
despatch
dispatch => Verbs of Sending and Carrying
route
forward => Verbs of Sending and Carrying
```

Example 2: Synset 01278717 includes Levin verbs which point to different classes. Since it consists of hyponym synsets (as indicated by the synset identifiers below), we go one level down in the taxonomy and classify the hyponym synsets separately.

```
push => Verbs of exerting force
jab poke 01296169 => Poke Verbs
nudge prod 00838894 => Verbs of contact
repel 01034588
shove 01278320   => Verbs of exerting force
ram 01296169
obtrude 01279473
thrust 01296169 => Verbs of exerting force
elbow shoulder 01278320
```

[7] No work on the latter is reported in this thesis; see, however, the discussion in section 6.3.1

[8] As we only consider first sense verbs here, for clarity, we refer to synsets in these examples as WordNet synset identifier codes, rather than their actual names. In addition, to simplify the examples somewhat, we refer to all Levin classes below as broad classes. In practice, the specificity of classification varies from class to class; see above section 6.3.1.

Example 3: Synset 00994853 includes thirteen member verbs, four of which are Levin “Verbs of Sending and Carrying”:

```
carry => Verbs of sending and carrying
port
airlift
lug => Verbs of sending and carrying
tote
chariot
bring  => Verbs of sending and carrying
ferry => Verbs of sending and carrying
church
tube
whisk
channel
retransmit
```

We need to classify more verbs to determine class assignment. To classify *whisk*, we extract its first sense from WordNet:

```
whisk - (move somewhere quickly; "The president was whisked away in his limo")
  => bring, convey, take - (take somebody or someone with oneself somewhere)
   => transport, carry - (move while supporting)
    => move, displace - (cause to move)
```

In LDOCE the verb has three senses. That corresponding to the WordNet first sense is identified as the second LDOCE sense shown below:

```
1. [T1] to move (something) quickly, exp. as to brush something off:
   "The horse was whisking its tail"
2. [X9 esp. OFF, AWAY] to remove
   a. by brushing lightly: "She whisked the dirt off"
   b. by taking suddenly: "She whisked the cups away / whisked him
      (off) home"
3. [T1] to beat (esp. eggs), esp. with WHISK.
```

The Levin classes already matched with the verbs in the same, hypernym and sister synsets are:

```
Verbs of putting
Verbs of removing
Verbs of sending and carrying
Verbs of exerting force
Poke verbs
Verbs of contact
Verb of cutting
Verbs of combining and attaching
Verbs of separating and disassembling
Verbs of throwing
Verbs of motion
```

From these classes, those whose syntactic description includes the LDOCE code X9 are:

```
Verbs of putting
Verbs of removing
Verbs of sending and carrying
Verbs of exerting force
Verbs of motion
```

After verifying these options manually, *whisk* is assigned to "Verbs of Sending and Carrying".

Example 4: The synset 01527059 includes around 90 member verbs related to the transfer of messages. These spread over nearly 60 hyponym synsets. Seven of the verbs are Levin verbs from various classes which include verbs taking sentential complements. Two of them are listed by Dorr (1997) as members of her new semantic classes. The synset is set aside for future work.

6.3.4 Completed Work

We applied the methods described in the above sections for (a) construction of semantic classes, (b) back-off estimates, and for (c) semantic classification of verbs as follows: using the semantic verb classification method described in the previous section, we analysed as exhaustively as possible three large WordNet verb files, assigning synsets in these files to semantic classes. The following three verb files were chosen because they covered most verbs used in our previous experiments:

- **35-verb.contact**. From the total of 513 synsets[9], 494 were classified as members of 17 broad[10] Levin classes. The classes are listed in table 6.7. 19 synsets were set aside for later classification.
- **38-verb.motion**. From the total of 888 synsets, 814 were assigned to the 23 Levin classes shown in table 6.8 and 71 synsets were left unclassified.
- **40-verb.possession**. From the total of 331 synsets, 273 were associated with the 10 Levin classes included in table 6.9. 58 synsets were left unclassified.

In addition, a small number of synsets (35) from other WordNet verb files were assigned to the Levin classes already listed in tables 6.7, 6.8, and 6.9, and to those of "Verbs of Assessment", "Verbs of Assuming a Position", and "Verbs of Concealment". Analysis of these synsets was a by-product of developing the approach. However, no further work was done on these other verb files.

From the total of 32 broad Levin classes exemplified among the classified WordNet synsets, 22 of the most frequent were chosen for further work. These were re-grouped

[9] Note that the total number of synsets refers here to the total number of synsets including verbs whose first sense belongs to the synset of question.

[10] Tables 6.7, 6.8 and 6.9 list only the broad Levin classes, not possible subclasses.

Levin Classes	**Classified Synsets**
9. Verbs of Putting	163
10. Verbs of Removing	32
12. Verbs of Exerting Force	5
13. Verbs Of Change of Possession	19
15. *Hold* and *Keep* Verbs	4
18. Verbs of Contact by Impact	54
20. Verbs of Contact	14
22. Verbs of Combining and Attaching	115
23. Verbs of Separating and Disassembling	13
24. Verbs of Colouring	3
35. Verbs of Searching	11
36. Verbs of Social Interaction	7
42. Verbs of Killing	27
44. Destroy Verbs	10
45. Verbs of Change of State	2
46. Lodge Verbs	1
47. Verbs of Existence	14

Table 6.7: Levin classes associated with WordNet "contact" verbs

Levin Classes	**Classified Synsets**
9. Verbs of Putting	66
10. Verbs of Removing	28
11. Verb of Sending and Carrying	44
12. Verbs of Exerting Force	22
17. Verbs of Throwing	39
18. Verbs of Contact by Impact	1
19. Poke Verbs	23
20. Verbs of Contact	2
21. Verbs of Cutting	53
22. Verbs of Combining and Attaching	4
23. Verbs of Separating and Disassembling	27
25. Verbs of Coloring	3
26. Verbs of Creation and Transformation	3
40. Verbs of Involving the Body	13
43. Verbs of Emission	8
44. Destroy Verbs	5
45. Verbs of Change of Possession	11
47. Verbs of Existence	51
48. Verbs of Appearance, Disappearance and Occurrence	6
49. Verbs of Body-Internal Motion	11
51. Verbs of Motion	383
53. Verbs of Lingering and Rushing	2
55. Aspectual Verbs	9

Table 6.8: Levin classes associated with WordNet "motion" verbs

Levin Classes	**Classified Synsets**
9. Verbs of Putting	8
10. Verbs of Removing	38
11. Verbs of Sending and Carrying	7
13. Verbs Of Change of Possession	156
15. Hold and Keep Verbs	8
25. Image Creation Verbs	15
29. Verbs with Predicative Complements	5
39. Verbs of Ingesting	11
47. Verbs of Existence	5
54. Measure Verbs	20

Table 6.9: Levin classes associated with WordNet "possession" verbs

Class Code	**Contains Levin Verbs of**	**Verbs for Back-off Estimates**
A	9. Putting	*place, lay, drop, load*
B	10. Removing: 10.1 - 10.3, 10.5 - 10.9	*remove, withdraw, steal, peel*
C	10. Removing: 10.4	*wipe, brush, filter*
D	11. Sending and Carrying 12. Exerting Force	*send, ship, carry* *push*
E	13. Change of Possession	*give, lend, contribute, donate, offer*
F	15. Hold and Keep 16. Concealment	*grasp, keep, store* *block, hide*
G	17. Throwing	*hit, throw, toss*
H	18. Contact by Impact 19. Poke Verbs	*bang, knock, punch* *pierce, poke*
I	20. Contact	*stroke, touch, kiss*
J	21. Cutting	*cut, clip, carve, chop, slice*
K	22. Combining and Attaching: 22.1 - 22.4	*add, mix, attach, lock*
L	22. Combining and Attaching: 22.5	*cling*
M	23. Separating and Disassembling: 21.3 - 23.3	*distinguish, tear, detach*
N	23. Separating and Disassembling: 23.4	*differ*
O	34. Assessment 35. Searching	*analyse, explore, investigate, survey* *fish*
P	36. Social Interaction	*communicate, marry, meet, visit*
Q	42. Killing	*kill, murder, strangle*
R	44. Destroy	*demolish, destroy, ruin*
S	47. Existence: Verbs of Spatial Configuration 50. Assuming Position	*hang, sit* *kneel, lie*
T	51. Motion	*arrive, move, slide, fly, sail*

Table 6.10: Semantic verb classes and verbs used for their back-off estimates

Class Code	Test Verbs
A	*cover, **drop**, **fill**, install, park, **place**, put, rearrange, set, space, superimpose*
B	*arrest, confiscate, dispel, exclude, exile, **remove**, rescue, save, **steal***
C	*shear*
D	***attract**, **bring**, **carry**, draw, hand, merchandise, **pull**, **send**,*
E	***acquire**, allocate, arm, **contribute**, credit, get, **give**, grant, letter, locate, obtain, **offer**, owe, pay, **provide**, receive, score, **supply**, win*
F	***keep**, **hide**, maintain, protect, reserve, retain, withhold*
G	*fire, **hit**, kick, single, **throw**, **toss***
H	*bump, hammer, **knock**, prick, rap, slam, slug, whip*
I	*neck, pet, **touch***
J	***carve**, hew, **slice***
K	***add**, **attach**, combine, **compare**, hook, join, mount, rejoin*
L	***cling***
M	***distinguish**, divide, segregate*
N	***differ***
O	***investigate**, probe, scan, seek*
P	***agree**, argue, bargain, compete, consult, fight, jest, **marry**, play, secede*
Q	-
R	***destroy**, eliminate*
S	***hang**, **kneel**, **lie**, lounge, orient, **sit**, stand*
T	*abandon, caper, **charge**, chase, coast, come, **dance**, drive, enter, flee, follow, go, haunt, head, hop, lead, leave, **move**, overhaul, pass, reach, return, run, **sail**, speed, **swing**, toe, turn, **walk***

Table 6.11: Classified test verbs

to our semantic classes by using the method described in section 6.3.1. This led to the combination of five pairs of broad Levin classes and the division of three into subclasses. The resulting 20 semantic classes are shown in table 6.10, labelled by class codes shown in the first column of the table. Back-off estimates for these classes were built using the method described in sections 6.3.2 and 5.2. The verbs used for obtaining the back-off estimates for each verb class are shown in the third column of table 6.10.

6.4 Experimental Evaluation

In this section we report the experimental evaluation of our refined and extended method to semantically motivated hypothesis selection. Section 6.4.1 introduces the test verbs employed and section 6.4.2 describes the SCF lexicons used in our experiments. Direct evaluation of the acquired lexicons is reported in section 6.4.3, task-based evaluation in the context of parsing in section 6.4.4.

6.4.1 Test Verbs

We selected for evaluation the same set of 474 test verbs as used by Carroll, Minnen and Briscoe (1998). 140 were found in the classified WordNet synsets. Our method assigned these verbs to semantic classes, as shown in table 6.11. As many as 118 are included in (Levin, 1993) and 106 in our annotated index, where they are classified according to their first sense. This big overlap is presumably due to both Levin's and Carroll, Minnen and Briscoe's selecting frequently occurring verbs as example/test verbs. This undoubtedly reduced the number of misclassifications our method made, as we assigned all 106 verbs (which occurred in the annotated Levin index) to semantic classes manually. However, among the remaining 34 non-annotated or non-Levin verbs, just one verb was classified incorrectly by our method: *locate* was associated with "Change of Possession" verbs (class E), while it should have been associated with "Verbs of Putting" (class A). This demonstrates that the semantic classification method is fairly accurate. The remaining 334 test verbs which were left unclassified, as they do not occur in any of the classified synsets, are listed in Appendix B.

6.4.2 Lexicons

We experimented with four different SCF lexicons. The data for these lexicons were obtained from 20 million words of BNC. Sentences containing an occurrence of one of the 474 test verbs were first extracted, on average of 1000 citations of each, and then processed using the SCF acquisition system's hypothesis generator. The parser employed in these experiments was a PCP (Chitrao and Grishman, 1990). Four different lexicons were constructed from the resulting SCF data using four different methods for hypothesis selection[11]:

1. **LEX-A:** Briscoe and Carroll's (1997) version of BHT
2. **LEX-B:** MLE thresholding
3. **LEX-C:** add-one smoothing and thresholding on smoothed estimates
4. **LEX-D:** linear interpolation with semantic back-off estimates for the 140 semantically classified verbs, and add-one smoothing for the 334 unclassified verbs, thresholding on smoothed estimates

When filtering the SCF data for LEX-D, any test verb which was used for constructing the back-off estimates was smoothed with a version of back-off estimates where this verb was excluded.

6.4.3 Evaluation of Acquired Lexicons

The acquired SCF lexicons were evaluated against a manual analysis of corpus data. The latter was obtained by analysing an average of 300 occurrences for each test

[11] These methods were introduced in earlier chapters. See section 2.5.3 for details of BHT and section 3.4.1 for those of MLE thresholding. Add-one smoothing and linear interpolation methods were described in section 5.3.

			System results				Unseen
Lexicon	KL	RC	Rank A. (%)	Precision (%)	Recall (%)	F	SCFs
LEX-A	0.55	0.67	72.4	55.3	49.4	52.2	75
LEX-B	0.55	0.67	63.8	84.5	47.2	60.6	75
LEX-C	0.56	0.72	65.2	86.9	51.8	64.9	0
LEX-D	0.29	0.88	78.3	87.1	71.2	78.4	4

Table 6.12: Average results for 45 semantically classified test verbs

			System results				Unseen
Lexicon	KL	RC	Rank A. (%)	Precision (%)	Recall (%)	F	SCFs
LEX-A	0.21	0.77	88.8	50.1	55.9	52.8	44
LEX-B	0.21	0.77	82.7	75.1	56.3	64.4	44
LEX-C	0.31	0.71	82.9	78.2	58.7	67.1	0

Table 6.13: Average results for 30 unclassified test verbs

verb in BNC or LOB, SUSANNE and SEC corpora. This evaluation was restricted to those test verbs for which the gold standard was readily available: 45 semantically classified and 30 unclassified verbs. These verbs are indicated in table 6.11 and in Appendix B in bold font. For these 75 verbs we calculated system results using type precision, type recall, ranking accuracy and F measure. We also calculated KL and RC between acquired unfiltered SCF distributions and gold standard distributions. The total number of SCFs unseen in the acquired SCF distributions which occurred in the gold standard distributions was also recorded. We did this to investigate how well the approach tackles the sparse data problem, i.e. the extent to which it is capable of detecting SCFs altogether missing in the data output by the hypothesis generator.

Table 6.12 gives average results for the 45 semantically classified test verbs in each lexicon. Those for the 30 unclassified test verbs in LEX-A, LEX-B and LEX-C are shown in table 6.13. In both tables the system results obtained with LEX-A (lexicon built using BHT) are clearly worse than those obtained with other lexicons. This shows on all measures except ranking accuracy. The ranking of SCFs is in fact identical in LEX-A and LEX-B - as indicated by RC - since for both lexicons, it is calculated using the MLEs straight from the SCF acquisition system's classifier. Ranking accuracy appears worse with lexicon LEX-B, however, because it only considers correct SCFs above the filtering threshold. With LEX-B there is a higher number of correct SCFs to consider and thus ranking accuracy shows worse results.

In both tables the system results obtained with LEX-C are better than those obtained with lexicon LEX-B. KL and RC results do not improve (except RC with semantically classified test verbs). This is for reasons discussed in chapter 5; add-one smoothing assigns identical probabilities/ranks to newly detected SCFs. Where it does so incorrectly, this shows only on KL and RC measures, which consider entire SCF distributions.

LEX-D is evaluated with the semantically classified test verbs only. The results included in table 6.12 show that the lexicon is clearly more accurate than the others examined. The improvement obtained with linear interpolation over the baseline MLE

Sem. Class	**Verbs Tested**	KL		RC		F Measure		Unseen SCFs	
		LEX-B	LEX-D	LEX-B	LEX-D	LEX-B	LEX-D	LEX-B	LEX-D
A	3	0.59	0.32	0.59	0.75	65.2	71.4	4	0
B	2	0.14	0.10	1.19	0.96	72.0	87.3	3	0
D	5	0.72	0.41	1.12	0.85	53.6	77.3	10	1
E	6	0.28	0.24	0.85	1.00	68.1	78.7	7	0
F	2	0.71	0.34	0.65	0.95	53.0	72.3	4	0
G	3	0.51	0.19	0.46	0.93	56.3	83.0	8	0
H	1	0.75	0.41	0.64	0.77	63.2	93.0	1	0
I	1	0.17	0.15	0.85	0.63	61.5	76.9	0	0
J	2	0.37	0.14	0.71	0.92	48.7	68.1	7	0
K	3	0.63	0.38	0.59	0.84	66.5	68.2	4	0
L	1	0.39	0.25	0.75	1.00	78.8	80.0	0	0
M	1	0.07	0.02	0.28	0.53	66.0	62.0	1	0
N	1	0.90	0.27	0.28	0.53	78.0	79.9	0	0
O	1	0.23	0.12	0.82	0.88	72.7	72.7	0	0
P	2	0.38	0.35	0.86	0.91	53.7	68.2	6	3
R	1	0.13	0.06	0.39	0.87	85.7	85.7	0	0
S	4	0.89	0.17	0.44	0.97	57.5	86.5	3	0
T	6	0.86	0.51	0.60	0.78	43.5	78.8	17	0

Table 6.14: Results for semantic classes

(LEX-B) is bigger than that reported in chapter 5. F measure improves here by 17.8, while in earlier experiments it improved by 7. The improvements obtained with RC and especially with KL are, moreover, clearly bigger with these experiments. Baseline results are lower than those in chapter 5, which leaves more room for improvement. They are worse probably because they use less data. In the earlier experiments, we used an average of 3000 citations of each test verb in corpus data, while here only 1000 were used. On the other hand, here we employed a refined method for constructing the semantic classes and back-off estimates and thus expect to see a bigger improvement in results.

From the total of 75 gold standard SCFs unseen in LEX-B only 4 are unseen in LEX-D. This indicates that the back-off estimates deal effectively with sparse data. Verb class specific results obtained with (i) the baseline MLE and (ii) linear interpolation methods allow us to examine the accuracy of the back-off estimates. These results are given in table 6.14. The table shows KL, RC and F measure results and the number of correct SCFs missing for (i) LEX-B and (ii) LEX-D. The first column shows a semantic verb class and the second indicates the number of verbs tested for the class. From a total of 20 classes, test verbs were found in 18. There are between one and six test verbs in each class. Thus the verb class specific results are not directly comparable, but serve to give us a general idea of the estimate's accuracy for each class.

KL, RC and F measure all agree that in 14 of the 18 verb classes, LEX-D outperforms LEX-B. KL shows improvement in all the 18 classes. RC indicates that when back-off estimates are used, the ranking of SCFs is better in all but one verb class. Class I (Levin verbs of "Contact") shows worse ranking with LEX-D than with LEX-B. This class is tested using one verb only: *touch*. This verb was used when constructing back-off estimates for the class. While testing, it was excluded, however, and the back-off estimates were constructed using two verbs only: *stroke* and *kiss*. Although these verbs take similar SCF options to *touch*, they rank them differently from *touch*.

With the F measure, LEX-D outperforms LEX-B in 15 classes. In one class, LEX-B outperforms LEX-D. This is class M, which includes Levin verbs of "Separating and Disassembling" (subclasses 23.1 - 23.3). Results are obtained using one test verb only: *distinguish*. Again, this verb was used in constructing the default set of back-off estimates for the verb class, but excluded when acquiring subcategorization for the verb itself. The back-off estimates employed here were thus constructed using only verbs *tear* and *detach*, which both take significantly fewer SCFs than *distinguish*. With two verb classes, the two lexicons show identical results. These are classes O and R, each tested with one verb only. No improvement was achieved because the empirically set (verb class specific) filtering thresholds appeared too high for these two individual verbs, resulting in too many false negatives.

All but two sets of back-off estimates tackled the sparse data problem efficiently. In LEX-D, there are gold standard SCFs missing only with verb classes D and P. The three SCFs unseen for class P (i.e. Levin verbs of "Social Interaction") occur with *agree*. The one SCF unseen for class D (i.e. Levin verbs of "Sending and Carrying" and "Exerting Force") occurs with *bring*. These two verbs are fairly polysemic and, in fact, the SCFs unseen involve senses not taken by the verbs used for back-off estimates.

6.4.4 Task-based Evaluation

The acquired SCF lexicons were also assessed using task-based evaluation in the context of parsing[12]. The idea was to examine the extent to which acquired SCF information improves accuracy of statistical parsing. This was done using the method proposed by Carroll, Minnen and Briscoe (1998). In the following, we shall first describe incorporation of the acquired SCF information into parsing and then give details of the evaluation.

Incorporating Acquired SCF Information into Parsing

The baseline non-lexicalised parsing system comprises[13]:

- an HMM POS tagger (Elworthy, 1994).
- an enhanced version of the GATE project lemmatizer (Minnen *et al.*, 2001).
- a wide-coverage unification-based phrasal grammar of English POS tags and punctuation.

[12] We are indebted to John Carroll for producing the parses and providing us with the GR data for evaluation.

[13] The tagger and grammar employed here are the same as used by Briscoe and Carroll's SCF acquisition system; see section 2.5.3.

AP
NONE
NP
NP_AP
NP_NP
NP_NP_SCOMP
NP_PP
NP_PPOF
NP_PP_PP
NP_SCOMP
NP_WHPP
PP
PP_AP
PP_PP
PP_SCOMP
PP_VPINF
PP_WHPP
PP_WHS
PP_WHVP
SCOMP
SINF
SING
SING_PP
VPBSE
VPINF
VPING
VPING_PP
VPPRT
WHPP

Table 6.15: VSUBCAT values in the grammar

- a generalized LR parser using Inui *et al.*'s (1997) variant of Briscoe and Carroll's (1993) statistical model, which uses the grammar, takes the results of the tagger as input and performs disambiguation.
- training and test treebanks (of 4600 and 500 sentences respectively) derived semi-automatically from the SUSANNE corpus.

The 500-sentence test corpus consists only of in-coverage sentences and contains a mix of written genres: news reportage (general and sports), *belles lettres*, biography, memoirs and scientific writing. The mean sentence length is 19.3 words (including punctuation tokens but excluding sentence-final full stop). It contains a total of 485 distinct verb lemmas and includes all verb types employed here as test verbs (see section 6.4.1).

In the experiment we took the four lexicons (from LEX-A to LEX-D) and assigned any SCF types missing (from the 163 possible) from these lexicons a probability using add-one smoothing. After this, the SCF probabilities in each acquired lexicon were factored into the parsing process during parse ranking at the end of the process. Complete derivations were ranked based on the product of (i) the (purely structural) derivation probability according to the probabilistic parse model and (ii) for each verb instance in the derivation, the probability of the verbal lexical entry that would be used in the particular analysis context. The entry was located via the VSUBCAT value assigned to the verb in the analysis by the immediately dominating verbal phrase structure rule in the grammar. This was possible as the VSUBCAT values are also present in the lexical entries acquired automatically using the same grammar. Table 6.15 lists the different VSUBCAT values. Some VSUBCAT values correspond to several of the 163 SCFs distinguished by the acquisition system. In these cases the sum of the probabilities of the corresponding lexical entries was used.

In taking the product of the derivation and SCF probabilities, some of the properties of a statistical language model are lost. The product is no longer strictly a probability, although it is not used here as such: it is used merely to rank competing analyses. Carroll, Minnen and Briscoe (1998) note that better integration of these two sets of probabilities is an area which requires further investigation.

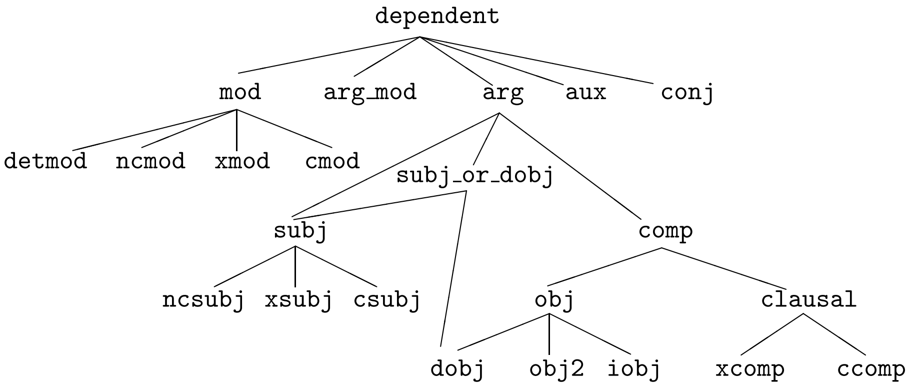

Figure 6.1: The grammatical relation hierarchy

Evaluation

Method The baseline and lexicalised parsers were evaluated against 500 test sentences marked up in accordance with a grammatical relation-based (GR) annotation scheme, described in detail by Carroll, Briscoe and Sanfilippo (1998) and Briscoe and Carroll (2000). This evaluation was chosen because it was found by Carroll, Minnen and Briscoe (1998) more sensitive to the argument/adjunct and attachment distinctions than the standard PARSEVAL bracketing evaluation they employed (Carroll *et al.*, 1997).

In general, grammatical relations (GRs) are viewed as specifying the syntactic dependency which holds between a head and a dependent. The GRs form a hierarchy, shown in figure 6.1. The most generic relation between a head and a dependent is *dependent*. Where the relationship between the two is known more precisely, relations further down the hierarchy are used. *Dependent* relations divide into *conj*(unction), *aux*(iliary), *arg*(ument), *mod*(ifier) and *arg_mod* relations. The latter relations refer to a semantic argument which is syntactically realised as a modifier (such as the passive *by*-phrase). *Mod*(ifier) relations divide further into determiner (*detmod*), non-clausal (*ncmod*), and clausal modifier relations controlled from within (*cmod*) or without (*xmod*). *Arg*(ument) relations divide initially into *comp*(lement), subject/object (*subj_or_obj*) and *subj*(ect) relations. *Subj*(ect) GRs divide further into clausal (*xsubj/csubj*) and non-clausal (*ncsubj*) relations. *Comp*(lement) GRs divide into *clausal* (*ccomp* controlled within and *xcomp* controlled without) and non-clausal *obj*(ect) relations. Below the latter, we still find the following relations: direct object (*dobj*), second (non-clausal) complement in ditransitive constructions (*obj2*), and indirect object complement introduced by a preposition (*iobj*).

In general the parser returns the most specific (leaf) relations in the GR hierarchy, except when it is unable to determine whether clausal subjects or objects are controlled from within or without (i.e. *csubj* vs. *xsubj*, and *ccomp* vs. *xcomp* respectively), in which case it returns *subj* or *clausal* as appropriate. Each relation is parameterised with a head (lemma) and a dependent (lemma), and optionally also with a type and/or specification of grammatical function. For example, the sentence (32a) would be marked up as in (32b).

(32) a *Paul intends to leave IBM.*
b *ncsubj (intend,Paul,_)*
xcomp (to,intend,leave)
ncsubj (leave,Paul,_)
dobj (leave,IBM,_)

When computing matches between the GRs produced by the parser and those in the corpus annotation, a single level of subsumption is allowed: a relation from the parser may be one level higher in the GR hierarchy than the correct relation. For example, if the parser returns *clausal*, this is taken to match both the more specific *xcomp* and *ccomp*. Also, an unspecified filler (_) for the type slot in the *iobj* and *clausal* relations successfully matches any specified filler. The head slot fillers are in all cases base forms of single head words: so for example, 'multi-component' heads, such as the names of people, places or organisations are reduced to one word.

(33) shows an example sentence from the test corpus:

(33) *They found deep pessimism in them.*

The GRs returned for this sentence by the baseline and lexicalised parsers are (34a) and (34b), respectively.

(34) a *ncsubj (find, they, _)*
dobj (find, pessimism, _)
ncmod (_, pessimism, deep)
iobj (in, find, they)
b *ncsubj (find, they, _)*
dobj (find, pessimism, _)
ncmod (_, pessimism, deep)
ncmod (in, find, they)

The latter is correct, but the former, incorrectly taking the PP to be an argument of *find*, gets penalised, receiving only 75% precision and recall.

Results Table 6.16 gives the result of evaluating the baseline and the lexicalised versions of the parser on the GRs annotation. It shows the results for the four lexicalised versions, obtained using the four sets of SCF probabilities from the different lexicons. The measures compare the set of GRs in the annotated test corpus with those returned by the parser, in terms of recall (the percentage of GRs correctly found by the parser out of all those in the treebank), precision (the percentage of GRs returned by the parser that are actually correct) and F measure. On these measures, the lexicalised versions show only slight improvement over the baseline parser. The results are mainly in accordance with those obtained with lexicon evaluation: the results with LEX-A are the worst while those with LEX-D are the best. However, the improvement obtained with LEX-D over the baseline is only 0.73 with F measure.

The results in table 6.16 are for all GRs. Results for argument GRs were closely

Method	**Precision (%)**	**Recall (%)**	**F**
Baseline parser	75.59	76.48	76.03
Lexicalised, LEX-A	76.06	77.14	76.59
Lexicalised, LEX-B	76.20	77.24	76.72
Lexicalised, LEX-C	76.18	77.26	76.71
Lexicalised, LEX-D	76.20	77.32	76.76

Table 6.16: GR evaluation for all GRs, before and after incorporation of SCF information

Method	**Precision (%)**	**Recall (%)**	**F**
Baseline parser	63.28	79.90	70.62
Lexicalised, LEX-A	70.48	74.75	72.55
Lexicalised, LEX-B	70.99	75.15	73.01
Lexicalised, LEX-C	70.94	74.95	72.89
Lexicalised, LEX-D	71.10	75.05	73.02

Table 6.17: GR evaluation for *comp* (lement) GRs only, before and after incorporation of SCF information

Method	**Precision (%)**	**Recall (%)**	**F**
Baseline parser	62.4	82.2	71.0
Lexicalised, LEX-D	71.7	76.4	73.9

Table 6.18: GR evaluation for *comp*(lement) GRs, before and after incorporation of SCF information from LEX-D. Only test sentences containing semantically classified test verbs are considered.

GR **Type**	**Baseline Parser**	**With Subcat** (LEX-D)	**Correct**
mod	472	525	21
ncmod	1995	2148	2434
xmod	24	46	129
cmod	139	140	208
detmod	1113	1114	1125
arg_mod	14	15	41
subj	24	22	1
ncsubj	1039	1039	1039
xsubj	0	0	5
csubj	5	6	3
obj	2	4	0
dobj	393	393	409
obj2	55	38	19
iobj	300	181	158
clausal	189	124	0
xcomp	260	262	323
ccomp	51	43	81
aux	376	370	379
conj	165	165	164
Total:	6616	6635	6539

Table 6.19: Numbers of each type of GR

similar. The lexicalised versions showed clearer improvements over the baseline parser, however, with complement GRs. This is illustrated in table 6.17, which shows results for complement GRs. The best results are obtained with LEX-D (only slightly better than with LEX-B), which improves 2.4 with F measure. Precision improves and recall worsens with each lexicalised version, as compared with the baseline results. With LEX-D, the 7.8% increase in precision is statistically significant even at the 99.9% level (*paired t-test*, T=6.17, 499 *df*). The 4.9% drop in recall is statistically significant at the 99% level (*paired t-test*, T=-3, 499 *df*).

Table 6.18 shows complement GR results for the baseline parser and the version lexicalised with LEX-D, for those 129 sentences which contain semantically classified verbs[14]. F measure for LEX-D now shows a 2.9 improvement over the baseline. Recall drops by 5.8% compared with the baseline, while precision increases by 9.3%. The increase in precision is again significant at the 99.9% level (*paired t-test*, T=3.73, 128 *df*). However, the drop in recall is no longer statistically significant at the 99% level, but only at the 95% level (*paired t-test*, T=-2.15, 128 *df*).

Table 6.19 gives the number of each type of GR returned by the baseline parser and when lexicalised with LEX-D, compared with the correct numbers in the test corpus. The baseline parser performs better than the lexicalised, when judged by the total number of GR returned, as opposed to the correct number in the test corpus. However,

[14] Recall that when constructing LEX-D, linear interpolation and semantically motivated back-off estimates were used for these verbs, while add-one smoothing was used for all other verbs.

the lexicalised parser is clearly better than the baseline when the total number of argument relations is considered (2318 are returned by the baseline parser, 2112 by the lexicalised parser, and 2038 occur in the test corpus), with complement relations contributing nearly all of this improvement.

Overall, the above results demonstrate that the scf probabilities can yield statistically significant improvements in parse accuracy. These are, however, insubstantial and mainly concern complement relations. Carroll, Minnen and Briscoe (1998) used the same evaluation method and test sentences to examine whether lexicalising the parser with scf frequencies acquired using Briscoe and Carroll's system (with bht for hypothesis selection) would improve parse accuracy. They reported 9.0% improvement in precision and 0.5% decrease in recall with argument relations. The improvement in precision was statistically significant, while the decrease in recall was not. In our experiment, bht (lex-a) did not yield statistically significant improvements even with complement relations. This is presumably because we employed a refined version of the parser and a more complete gr annotation scheme.

With this task-based evaluation the differences in results between lexicons are not as great as we would expect on the basis of the lexicon evaluation reported in the previous section. One reason for this could lie in the evaluation method: the approach of combining the probabilities of several scfs to obtain a probability for a single vsubcat value. Essentially, this approach involves "reducing" the 163 scfs into 29 vsubcat values. Not only are many scf distinctions lost in doing this, but the approach can also alter the ranking of scfs for verbs. For example, it is possible that the resulting highest ranked vsubcat value for a verb may not correspond to the vsubcat value of the highest ranked scf for this verb. For *keep* in our gold standard, for example, the value np will become the highest ranked, although np-pp is the value of the highest ranked scf.

To investigate this effect, we considered the 129 test sentences which contain semantically classified verbs and for each test verb manually examined how much it affects the parse ranking if, instead of the probability of vsubcat value, we consider the probability of the vsubcat value of the scf in question. Contrary to what we had expected, this had virtually no effect on results. While examining the test sentences manually we noticed, however, that many scfs seemed "typical" for the verbs they occurred with. When we considered the 45 test verbs for which we had manually analysed (gold standard) corpus data, we noticed that, from the total of 77 occurrences of these verbs in the 129 test sentences, 40% were with the scf ranked the highest in the gold standard and 37% were with the scf ranked the second or third highest. For instance, *hit* occurred twice in our test data, and both times with scf np which, according to our gold standard, is its highest ranked frame. Thus according to the gold standard, in 77% of cases a high frequency scf was evaluated for a verb.

As we did not have a gold standard for all the 474 test verbs employed, we could not extend this investigation to the entire test data. On the basis of this smaller investigation it seems, however, that the 500 sentence test data employed are not adequate for comparing the scf frequencies between the lexicons examined. There is very little difference in accuracy between the various lexicons with the highest ranked scfs. Back in section 3.4.2 we showed, for example, that despite its poor overall performance, BHT nearly always acquires the three most frequent SCFs of verbs correctly. The MLE, add-one smoothing and linear interpolation methods likewise perform well with high frequency SCFs. Thus properly to compare the different lexicons using this evaluation method, we would need test sentences which exemplify a higher number of medium and low frequency SCFs for the verbs tested.

6.5 Discussion

Direct evaluation of acquired lexicons showed that the approach to semantically motivated SCF acquisition can yield significant improvements when applied to large-scale lexical acquisition. At best, it achieved 78.4 F measure with 45 test verbs. On the same set of verbs, Briscoe and Carroll's original BHT method achieved 52.2 F measure, and the baseline MLE method 60.6 F measure. Our result compares favourably also with results obtained with the other equally ambitious SCF acquisition systems discussed in chapter 2.

Task-based evaluation showed that the SCF probabilities acquired using our method can improve the parse accuracy of a statistical parser. The improvements obtained were not considerable; however, they were statistically significant when the evaluation was restricted to complement GRs and to sentences which contained verbs for which subcategorization probabilities were acquired using the semantically motivated method for hypothesis selection.

The semantically motivated method could be extended and improved in several ways. Extensions are required before the approach can be used to benefit the entire lexicon. Firstly, a comprehensive set of semantic classes and back-off estimates is needed. This requires refinement and extension of Levin classification. As discussed in section 6.3.1, we can approach the task by building on previous work, e.g. on refined Levin classes by Dang *et al.* (1998), the new semantic classes proposed by Dorr (1997) and the new diathesis alternations by Korhonen (see Appendix C). Secondly, the semantic classification of WordNet synsets needs to be completed. We covered most synsets in three large WordNet verb files; however, further work is required on the 148 synsets left unclassified in these files, and on the synsets in the remaining 12 WordNet verb files.

Refinements are required in the current approach to back-off estimates. For some verb classes, back-off estimates were constructed using fewer verbs than the ideal 4-5, because not enough Levin verbs were found in the annotated index that would occur frequently enough in the corpus data. In the lexicon evaluation, these estimates proved insufficient for some test verbs. We could address this problem by using also non-Levin verbs for back-off estimates. For example, the verbs correctly assigned to semantic classes by our classification method could be considered as candidates.

In lexicon evaluation, the semantic classes employed proved fairly distinctive in terms of subcategorization. Accuracy could further be improved by narrowing down the current classes into more specific (sub)classes, where possible. This would, of course, increase the manual effort involved in the approach, as each novel class requires manually constructed back-off estimates. We could investigate the possibility of constructing back-off estimates automatically or semi-automatically. One idea would be to use the SCF acquisition system to hypothesise the SCF distributions needed for back-off estimates. If this were not to yield accurate enough estimates, one could manually verify the automatically acquired distributions and remove any incorrect SCF assignments. Further research is needed to determine how well this approach would work in practise.

Narrowing down the semantic classes would be especially helpful if, instead of polysemic SCF distributions, we were concerned with verb sense specific SCF distributions. In future, the system's hypothesis generator could be modified to hypothesise such distributions, using WSD techniques on the predicate forms. This would reduce noise in hypothesis selection and in the subcategorization acquisition process in general. For this, the verb classification algorithm would also require modification. It currently assigns verbs to semantic classes according to their first sense only.

Currently, we deal with the problem of polysemy by assigning predicates to semantic classes corresponding to their predominant sense. An easy way of improving this approach would be to assign predicates to classes corresponding to all their senses. We could thus obtain back-off estimates for a polysemic predicate by merging the back-off estimates of all its semantic classes. The contribution of each set of back-off estimates could be weighted according to the frequency of the respective sense in WordNet. This would allow detection of those SCFs related to less frequent senses, while still giving most weight to the back-off estimates of the predominant sense. Although it is clear that modifying the system's hypothesis generator to hypothesise verb sense specific SCF distributions (as discussed above) is a better long term solution to the problem of polysemy, this approach would offer a quick way of improving the extant approach.

Lexicon evaluation showed that the semantically motivated method yields significant improvements in hypothesis selection. It is especially efficient in addressing the problem of low frequency data discussed in chapter 3. With evaluation on 45 test verbs, the method achieved 87.1% precision, while the baseline MLE method also achieved impressive 84.5% precision. The crucial difference between the two methods showed up in recall. This was 71.2% for the semantically motivated method and only 47.2% for the MLE method. As filtering in both methods is based on cutting off the low frequency data, the 24% improvement in recall is due to more appropriate handling of sparse data. However, the approach could be further improved. Currently, the verb class specific filtering threshold is established empirically, using held-out training data. Our evaluation revealed that this does not deal optimally with the variations in the number of SCFs taken by individual verbs. Rather, too few/many SCFs are accepted for some verbs. A way to address this problem would be to weight the empirically defined threshold by the number of SCF options for an individual verb in a dictionary such as ANLT or COMLEX.

6.6 Summary

In this chapter, we first discussed earlier work on semantically motivated lexical acquisition and then outlined our own approach. Essentially, we adopted the new approach to hypothesis selection proposed in chapter 5, refined it further and modified it for large-scale SCF acquisition. The resulting overall approach involves automatically assigning verbs to semantic classes on the basis of their most frequent sense. This is done by choosing the semantic class already associated with the respective WordNet synset. Hypothesis selection is conducted by ranking the hypothesised SCFs according to their MLEs, by smoothing the conditional distribution with back-off estimates of the respective verb class, and by setting an empirically defined threshold on the resulting estimates to filter out unreliable SCFs.

We evaluated our semantically motivated approach with unknown test verbs using two methods: direct evaluation of the acquired lexicons and task-based evaluation in the context of parsing. The approach was compared with three other approaches to hypothesis selection (the BHT, MLE thresholding and add-one smoothing methods). Lexicon evaluation showed that our method yields subcategorization information significantly more accurate than that obtained by the other methods examined. Task-based evaluation showed that the subcategorization probabilities acquired by our method can improve the performance of a statistical parser. With task-based evaluation, there were no substantial differences between the various methods of hypothesis selection; rather, the semantically motivated approach achieved only slightly better results than the other methods. We discussed possible reasons for this.

Finally, we discussed ways in which the proposed method could be further refined. We also considered the modifications and extensions required successfully to apply the method across the entire lexicon.

Chapter 7

Conclusions

In this concluding chapter, we summarise the contributions of this thesis (section 7.1) and outline directions for future research (section 7.2).

7.1 Contributions of this Thesis

The main contribution of this thesis was to improve the accuracy of automatic subcategorization acquisition. We did this by improving the critical hypothesis selection phase of subcategorization acquisition, reported to be the weak link of many SCF acquisition systems. Our work resulted in various experimental findings and methodological proposals which we summarise as follows.

I Hypothesis Testing We addressed the widely-recognized problem that statistical filtering - used by most systems to remove noise from automatically acquired SCFs - performs badly, especially with low frequency data. We conducted experiments where we compared three different approaches to hypothesis selection. These were (i) a filter based on the binomial hypothesis test, (ii) a filter based on the binomial log-likelihood ratio test, and (iii) a filter which uses a threshold on MLEs of SCFs from the hypothesis generator. Surprisingly, the simple MLE thresholding filter worked best. The BHT and LLR both produced an astounding number of FPs, particularly at low frequencies. Our investigation showed that hypothesis testing does not work well in subcategorization acquisition because not only is the underlying distribution zipfian but nor is there significant correlation between conditional and unconditional SCF distributions. The lack of correlation between the two distributions also affects refinements of MLE thresholding such as smoothing or Bayesian estimation. Thus more accurate back-off estimates are needed for SCF acquisition than those assumed so far, especially if we are to deal effectively with low frequency data.

II Back-off Estimates Assuming that unconditional SCF distribution provides accurate back-off estimates for all verbs is equivalent to assuming that all verbs behave uniformly with respect to subcategorization. We pointed out that this assumption is in contradiction with simple observations about verb behaviour, as well as with linguistic research, which has shown that it is possible to associate verbs with semantically and syntactically motivated classes that capture subcategorization behaviour characteristic of their members.

We examined experimentally to what extent we could exploit linguistic verb classifications in automatic subcategorization acquisition. We did this by experimenting with a set of SCF distributions specific to verb form (as opposed to verb sense). Employing the semantic verb classification of Levin (1993) and the syntactic classification obtained from the ANLT dictionary, we examined to what extent verbs classified similarly in these resources correlate in terms of SCF distributions. The results showed that the degree of SCF correlation was greater with semantically and syntactically similar verbs than with all verbs in general, and that the correlation between semantically similar verbs was better than that between syntactically similar verbs. The best SCF correlation was observed when verbs were classified semantically according to their predominant sense. These results suggest that more accurate back-off estimates can be obtained for SCF acquisition by exploiting generalizations from linguistic theory. On the basis of our results, we proposed assigning verbs to semantic classes matching their predominant sense and obtaining back-off estimates specific to these classes ($p(scf|class)$).

III Semantically Motivated Hypothesis Selection We presented a novel approach to hypothesis selection suitable for large scale SCF acquisition which uses semantically motivated back-off estimates. This approach makes no use of statistical hypothesis tests. Instead, it builds on MLE thresholding.

- **Semantic Classes** Our semantic classes were based on Levin's. As it was important to minimise the cost involved in constructing back-off estimates, we did not adopt all Levin classes as they stand, although this would have allowed maximal accuracy. Rather, a method was devised for determining the specificity of the Levin class(es) required for reasonable distinctiveness in terms of subcategorization. This method involves examining the similarity between Dorr's (1997) LDOCE codes for Levin classes and the SCF similarity between verbs in these classes. While Levin proposed altogether 48 broad (191 specific) classes for verbs taking NP and PP complements, we estimated that not more than 50 classes are required across the entire lexicon. We applied the method to 22 broad Levin classes, which resulted in 20 semantic classes.

- **Verb Classification** A technique was devised which automatically assigns verbs to semantic classes via WordNet (Miller *et al.*, 1993). Levin's verb index was annotated for verbs classified according to their predominant sense, and a semi-automatic algorithm was designed which assigns WordNet synsets to semantic classes. This algorithm makes use of the annotated Levin index, the LDOCE dictionary and Dorr's LDOCE codes for Levin classes. We applied the algorithm to the total of 1616 synsets. Given the resulting synset-class associations, individual verbs are automatically classified according to the semantic class associated with the synset corresponding to their first sense. Our technique exploits some ideas from Dorr (1997). However, it differs from Dorr's technique in several ways which contribute to increased accuracy, classifying verbs according to their predominant sense and building a more static lexical resource.

- **Back-off Estimates for Semantic Classes** A method was proposed for constructing back-off estimates based on semantic verb classes. This involved choosing from each class (ideally) 4-5 verbs whose predominant sense corresponds to the class, manually analysing corpus data to obtain SCFs distributions for these verbs, and automatically merging the resulting distributions to obtain back-off estimates for the class. Using this method, we constructed back-off estimates for 20 semantic classes.

- **Filtering** The proposed semantic verb classes, verb classification technique and back-off estimates could be used in hypothesis selection in various ways. Our novel approach involves (i) identifying the semantic class of a predicate, (ii) acquiring a conditional SCF distribution for the predicate from corpus data (using a subcategorization acquisition system's hypothesis generator), (iii) smoothing the conditional distribution with back-off estimates of the semantic class of the predicate using linear interpolation, and (vi) setting an empirically defined threshold on the probability estimates from smoothing to filter out unreliable hypotheses.

According to the evaluation reported, this semantically motivated approach to hypothesis selection provides an effective way of dealing with sparse data. It yields subcategorization information substantially more accurate than that obtained by baseline MLE thresholding. When we performed experiments where we smoothed the conditional SCF distributions of predicates with the unconditional distribution, this yielded subcategorization information less accurate than that obtained by MLE thresholding. This shows that poor back-off estimates are worse than none (Gale and Church, 1990). Overall, our result demonstrates that it is beneficial to guide subcategorization acquisition with a priori semantic knowledge. Such knowledge allows detection of SCF information that does not emerge from purely syntactic analysis.

The experimental findings and various methods proposed in this thesis contribute directly to the development of subcategorization acquisition and thus to obtaining more accurate subcategorization lexicons. The work reported in this thesis is potentially of interest to many practical NLP applications which use subcategorization information, and to linguistic theory. Knowledge about novel verb and verb class associations, and accurate SCF and verb associations can also be used to test and enrich linguistic theory (e.g. Levin, 1993).

7.2 Directions for Future Research

7.2.1 Hypothesis Selection

Further work is required before the novel approach can be applied to benefit the entire lexicon. Firstly, a comprehensive set of semantic classes and back-off estimates is needed which covers the entire lexicon. This will require refining the Levin classification (e.g. Dang *et al.*, 1998) and extending it to cover novel verb classes (e.g. Dorr, 1997). Secondly, the semantic classification of WordNet synsets needs to be completed so that verbs across the entire lexicon can automatically be assigned to semantic classes.

The proposed approach can also be improved in several respects. Many of the semantic classes employed could be narrowed down further to obtain clearer subcategorization distinctions. We opted for fairly general classification due to the high cost involved in obtaining back-off estimates. However, it would be worth investigating ways of reducing this cost. One option would be to obtain the conditional distributions needed for back-off estimates automatically using the subcategorization acquisition machinery with MLE thresholding for hypothesis selection. If this does not yield accurate enough estimates - which is likely, given the poor performance of MLE thresholding with low frequency data - we could examine whether it would help manually to verify the automatically acquired distributions and remove incorrect SCF assignments.

Obviously, one could also reduce the cost involved in obtaining back-off estimates by reducing the number of conditional distributions used for these estimates. Our evaluation showed, however, that when fewer distributions than the ideal 4-5 were used (due to the lack of suitable Levin verbs), back-off estimates did not turn out comprehensive enough. Thus this is not an ideal solution. Rather, it seems sensible also to consider non-Levin verbs (e.g. verbs from semantically classified synsets) as candidates to obtain the ideal number of conditional distributions for back-off estimates, and consider ways of reducing the cost in constructing estimates.

According to our evaluation, when back-off estimates were accurate, they helped to deal effectively with sparse data. However, the benefits from smoothing did not always show up in the acquired lexicon because the empirically defined filtering threshold (specific to verb class) either accepted too many or too few SCFs. It is worth investigating ways of more accurate thresholding. One option would be to weight the empirically defined threshold by the number of SCF options for an individual verb in a dictionary such as ANLT or/and COMLEX.

Currently we deal with the problem of polysemy by assigning predicates to semantic classes corresponding to their predominant sense. A better approach might be to consider all senses of predicates and allow for multiple class assignment. For a polysemic predicate, back-off estimates could be obtained by merging the back-off estimates of all its semantic classes. When doing this, we could weight the contribution of each set of back-off estimates according to the frequency of the sense in WordNet. In other words, the back-off estimates of the predominant sense would still have the biggest effect, but the other estimates would allow detection of those SCFs merely related to less frequent senses.

A more effective way to deal with polysemy would be to modify the subcategorization acquisition system to hypothesise SCF distributions specific to verb sense by using WSD techniques on the predicate forms[1]. This would allow us to assign occurrences of predicates to semantic classes corresponding to their appropriate senses and thus reduce noise in hypothesis selection and subcategorization acquisition in general. To gain full benefit from this approach, we would need to narrow down semantic verb classification, obtain back-off estimates specific to finer-grained classes, investigate ways of reducing the cost of obtaining back-off estimates and refine the verb classification algorithm so that it is capable of allocating predicate senses to semantic classes.

The starting point in this thesis was to investigate why hypothesis tests have been reported to perform poorly in subcategorization acquisition. While we detected reasons for this, we did not further refine the filters based on hypothesis tests. It would be interesting to integrate semantically motivated back-off estimates into hypothesis tests and investigate to what extent this improves performance.

7.2.2 Hypothesis Generation

The work reported in this thesis has concentrated on improving the hypothesis selection phase of subcategorization acquisition. There is, however, a limit to how far we can get by merely refining hypothesis selection. To render full subcategorization recovery possible, improvements are required in hypothesis generation as well.

One way of improving hypothesis generation would be to enhance parse selection accuracy. In this thesis, we demonstrated that using the SCFs acquired by the system to rerank analyses returned by the parser can improve parser accuracy. Incrementally integrating this and other lexical information into probabilistic parsing could help.

Current subcategorization acquisition systems generate hypotheses about associations of predicates with SCFs. However, there is more to subcategorization than syntactic frames; the entire range of phenomena we discussed in chapter 2: linking between syntactic and semantic levels of predicate-argument structure, semantic selectional restrictions/preferences on arguments, control of understood arguments in predicative complements, diathesis alternations and so forth. The eventual aim is to supplement a target lexicon with all this information. Knowledge of these further details of subcategorization will also aid hypothesis generation. The various components of argument structure are interrelated, so that knowledge about one component can aid the automatic recovery of another.

For example, if a subcategorization acquisition system gathers from corpus data information about lemmas which occur as heads of arguments in SCFs given predicates, this information could be used as input to predicate sense disambiguation. This would allow hypothesising associations between verb senses and SCFs. The argument head data could also be used as input to selectional preference acquisition (e.g. McCarthy, 2001). Knowledge about selectional preference(s) of predicates on their arguments would help to disambiguate argument senses (e.g. Kilgarriff and Rosenzweig, 2000).

[1] Some work already exists on this, see section 7.2.3.

In addition, as Briscoe and Carroll (1997) and Boguraev and Briscoe (1987) suggest, the ability to recognize that argument slots of different SCFs for the same predicate share semantic restrictions/preferences would assist recognition that the predicate undergoes specific diathesis alternations. This in turn would assist inferences e.g. about control, equi and raising, which again would help to narrow down some SCF options.

Similarly, knowledge about alternations would help to distinguish e.g. between unergative, unaccusative and object-drop verb types (Levin, 1993). These verbs take similar intransitive and transitive SCFs, but assign different thematic roles to their subject and object arguments in the event described:

(35) a **Unergative**:
The plane flew to Rome ↔ *Bill flew the plane to Rome*
NP_{agent} flew ↔ NP_{causer} flew NP_{theme}

b **Unaccusative**:
Snow melted in the kettle ↔ *They melted snow in the kettle*
NP_{theme} melted ↔ NP_{causer} melted NP_{theme}

c **Object-drop**:
Mary ate the food ↔ *Mary ate*
NP_{agent} ate NP_{theme} ↔ NP_{agent} ate

Furthermore, classifying verbs semantically according to their alternation behaviour would aid prediction of unseen SCF behaviour and induction of low frequency frame associations (as demonstrated with hypothesis selection in this thesis). If frequency of alternations were known, these predictions could be made via statistical estimation of the semi-productivity of alternation rules (Briscoe and Copestake, 1999).

7.2.3 Extending the Scope

Work is under way to develop subcategorization acquisition in the directions discussed above. To provide an idea of the state of the art, we shall now give a brief overview of such work undertaken recently around Briscoe and Carroll's system. We contributed to some of this work while working on the research reported in this thesis.

McCarthy (2001) has developed a system which acquires selectional preferences from a SCF lexicon extracted using Briscoe and Carroll's system. The system uses the list of head lemmas in argument slots in SCFs and the WordNet semantic noun hierarchy to infer a probability distribution on semantic classes occurring in a given argument position in a given SCF for specific predicates. This probability distribution characterizes the selectional preference(s) of the predicate on that argument.

The technique employed for selectional preference acquisition is based on that proposed by Li and Abe (1995). The preferences for a slot are represented as a tree cut model (TCM). This is a set of classes which cuts across the WordNet noun hypernym hierarchy covering all leaves disjointly. The argument head data is collected from a slot and used to populate the WordNet hierarchy with frequency information. Each head lemma is assigned a WordNet class where it occurs as a synonym. If a lemma is ambiguous between classes, then counts are evenly distributed between these classes.

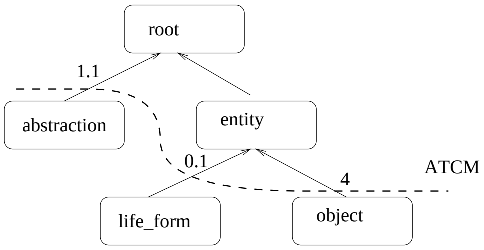

Figure 7.1: TCM for *build* Object slot

The frequency at each class is then propagated up the IS-A links of the hierarchy so that the frequency counts from hyponym classes are added to the count for each hypernym class. A root node contains the total frequency count for all argument head lemmas that were found in WordNet. The appropriate level of generalization - the best TCM - is determined using the Minimum Description Length (MDL) principle (Rissanen, 1978). The MDL principle finds the set of classes that make the best compromise between a good fit for the data and providing a succinct model. Figure 7.1 displays, as an example, how part of the TCM for the direct object slot of *build* might look[2].

McCarthy has used the selectional preference distributions for WSD on nouns occurring in SCF slots with results of around 70% precision (Kilgarriff and Rozenweig, 2000) and also to identify whether a specific predicate participates in a diathesis alternation (McCarthy and Korhonen, 1998; McCarthy, 2001). Identifying participation in alternations on purely semantic grounds would be difficult, as the subtle lexical-semantic components which give rise to alternations are not easily defined (e.g. Levin and Rappaport, 1996) and as alternations are semi-productive in nature (e.g. Briscoe and Copestake, 1999). Looking for near-identical selectional preference distributions on argument slots between putatively alternating SCFs is an alternative option. McCarthy's method is suitable for detecting participation in alternations where a particular argument type appears in slots which have different grammatical roles in the alternating frames. One example is the causative-inchoative alternation, where the object of the transitive variant can also appear as the subject of the intransitive variant:

(36) *The boy broke* ***the window*** ↔ ***The window*** *broke*

McCarthy first uses syntactic processing to find candidate verbs taking the alternating SCFs. For this, a SCF lexicon acquired using Briscoe and Carroll's system is screened for candidate verbs which occur with the SCFs involved in an alternation. The latter are obtained from a mapping which links the SCFs involved in Levin alternations to those recognized by Briscoe and Carroll's system[3]. Selectional preferences are then ac-

[2] The type of TCM exemplified in this figure is an Association TCM. See McCarthy and Korhonen (1998) and McCarthy (2001) for further details of this and other TCM types employed.

[3] We contributed to McCarthy's work by producing this mapping. A brief description of the mapping is included in Appendix C.

quired for the slots involved in an alternation using as input data the argument heads stored in the lexical entries. Verbs which participate in alternations are expected to show a higher degree of similarity between the preferences at the target slots compared with non-participating verbs. To compare preferences, probability distributions across WordNet are compared using a measure of distributional similarity. McCarthy (2001) reported a significant relationship between similarity of selectional preferences at alternating slots, and participation in the causative and conative alternations. At best, 72% accuracy (against 50% baseline) was obtained for the causative alternation using euclidean distance (Lee, 1999) as the measure of distributional similarity.

Although this is a promising result, applying the method to a wider range of alternations will largely depend on overcoming the sparse data problem. Many alternations involve rare verbs and for many verbs that participate in alternations, one of the alternating forms is rare. This problem could partly be addressed by improving the accuracy of subcategorization acquisition, e.g. by using the novel method for hypothesis selection proposed in this thesis.

Recently McCarthy has worked on disambiguating verb forms into WordNet senses using the distribution of argument heads in argument slots[4]. If this work proves successful, it should be possible to apply the techniques discussed in this and above sections to predicate senses directly, and thus reduce noise in SCF, selectional preference and alternation acquisition.

Most techniques discussed in this section require further development before they can be integrated into subcategorization acquisition machinery to benefit large-scale hypothesis generation. In the meantime, the novel semantically-driven approach to hypothesis selection proposed in this thesis allows us to some extent to compensate for the semantic information currently missing in hypothesis generation.

[4] This work has not yet been published.

Appendix A

SCF Classification

A.1 Introduction

The below list details the 163 SCFs employed by Briscoe and Carroll's SCF acquisition system. The SCFs were constructed by manually merging the SCFs of the ANLT and COMLEX syntax dictionaries and adding around 30 SCFs found by examining unclassifiable patterns of corpus examples. These consisted of some extra patterns for phrasal verbs with complex complementation and flexible ordering of the preposition or particle, some for non-passivizable patterns with a surface direct object, and some for rarer combinations of governed preposition and complementizer combinations. The resulting SCFs abstract over specific lexically-governed particles and prepositions and specific predicate selectional preferences but include some derived semi-predictable bounded dependency constructions, such as particle and dative movement.

(37) shows a legend for a SCF entry in the classification. The first line shows the COMLEX SCF name (for the first 116 SCFs which appear in COMLEX). It also indicates the frequency of the SCF in ANLT. Where this is 0, the SCF does not appear in ANLT. Cases marked as '??' are unsure. The second line gives the frame specification using ANLT notation (for the last 47 SCFs - which do not appear in COMLEX but only in ANLT and/or corpus data - this specification is given in the first line of the entry) and, for some SCFs, the mapping to an XTAG tree family. The following line shows a tagged example sentence from corpus data where the SCF occurs. The final line gives the SCF specification according to the grammar employed by Briscoe and Carroll's system. It indicates the tag sequence grammar (TSG10vs) feature values and headwords in parse trees. For full details of the classification and the mapping between ANLT and COMLEX, see Briscoe (2000).

(37)
```
SCF class number. COMLEX class name / Frequency of the class in ANLT
(ANLT SUBCAT/SUBTYPE PFORM/PRT feature value pairs) / XTAG :Tree-family
Example sentence with CLAWS-II/Susanne TAGS
TSG10vs PSUBCAT/VSUBCAT, PRT, PFORM and headwords/HEADWORDS | *_TAGS
```

A.2 Classification

```
   1. ADJP / 93
   (SUBCAT SC_AP, SUBTYPE EQUI) / XTAG: Tnx0Va1
his_AT reputation_NN1 sank_VVD low_JJ
   (VSUBCAT AP)

   2. ADJP-PRED-RS / 15
   (SUBCAT SC_AP, SUBTYPE RAIS) / XTAG:Tnx0Ax1
he_NP1 appears_VVZ crazy_JJ / distressed_VVN
   (VSUBCAT AP) / (VSUBCAT VPPRT)

   3. ADVP / 64
   (SUBCAT ADVP)
he_NP1 meant_VVD well_RP
   (VSUBCAT NONE, PRT +) well

   4. ADVP-PRED-RS / 0 (in vppp)
   (SUBCAT ADVP, SUBTYPE RAIS)
He_NP1 seems_VVZ well_RP
   (VSUBCAT NONE, PRT +) well

   5. AS-NP / 0 (in vppp with PRT 1 = end)
   (SUBCAT SC_NP, SUBTYPE EQUI, PREP as)
I_NP1 worked_VVZ as_CSA an_AT1 apprentice_NN1 cook_NN1
   (VSUBCAT PP) as

   6. EXTRAP-NP-S / 58
   (SUBCAT NP_SFIN, SUBTYPE EXTRAP, AGR N2[NFORM IT])
it_PPH1 annoys_VVZ them_PPHO2 that_CST she_PPHS1 left_VVD
   it (VSUBCAT NP_SCOMP) * *_VVZ/D/G

   7. S-SUBJ-NP-OBJ / 58
   (SUBCAT NP_SFIN, SUBTYPE EXTRAP, AGR S[FIN +]) / XTAG:Ts0Vnx1
that_CST she_PPHS1 left_VVD annoys_VVZ them_PPHO2
   *_VVD/Z/G (VSUBCAT NP)

   8. TO-INF-SUBJ-NP-OBJ / 56
   (SUBCAT OC_INF, SUBTYPE EQU_EXTRAP, AGR VP[FIN -])
to_TO read_VV0 pleases_VVZ them_PPHO2
   *_VV0 (VSUBCAT NP)

   9. EXTRAP-TO-INF / 4
   (SUBCAT VPINF, SUBTYPE EXTRAP, AGR N2[NFORM IT])
it_PPH1 remains_VVZ to_TO find_VV0 a_AT1 cure_NN1
   IT (VSUBCAT VPINF)

   10. EXTRAP-FOR-TO-INF / 0 (not in vppp)
   (SUBCAT SINF, SUBTYPE EXTRAP, AGR N2[NFORM IT])
it_PPH1 remains_VVZ for_IF us_PPHO2 to_TO find_VV0 a_AT1 cure_NN1
   IT (VSUBCAT PP_VPINF) for (PSUBCAT NP)

   11. EXTRAP-NP-TO-INF / 56
   (SUBCAT OC_INF, SUBTYPE EQU_EXTRAP, AGR N2[NFORM IT])
it_PPH1 pleases_VVZ them_PPHO2 to_TO find_VV0 a_AT1 cure_NN1
   IT (VSUBCAT SINF)
   12. EXTRAP-TO-NP-S / 5  (4 without EXTRAP)
   (SUBCAT PP_SFIN, SUBTYPE EXTRAP, PFORM to, AGR N2[NFORM IT])
it_PPH1 matters_VVZ to_II them_PPHO2 that_CST she_PPHS1 left_VVD
   IT (VSUBCAT PP_SCOMP) to (PSUBCAT NP)   *_VVZ/D/G

   13. EXTRAP-TO-NP-TO-INF / 1
   (SUBCAT PP_VPINF, SUBTYPE EXTRAP, PFORM to)
it_PPH1 occurred_VVD to_II them_PPHO2 to_TO watch_VV0
   IT (VSUBCAT PP_VPINF) to (PSUBCAT NP)

   14. S-SUBJ-TO-NP-OBJ / 5
   (SUBCAT PP_SFIN, SUBTYPE EXTRAP, AGR S[FIN +])
that_CST she_PPHS1 left_VVD matters_VVZ to_II them_PPHO2
   *_VVD/G/Z (VSUBCAT PP) to (PSUBCAT NP)

   15. FOR-TO-INF / 17
   (SUBCAT SINF)
I_PPHS1 prefer_VV0 for_IF her_PPHO1 to_TO do_VV0 it_PPH1
   (VSUBCAT PP_VPINF) FOR (PSUBCAT NP)

   16. HOW-S / 155 (combined with other wh comps)
   (SUBCAT WHS)
he_PPHS1 asked_VVD how_RGQ she_PPHS1 did_VDD it_PPH1
   (VSUBCAT PP)  HOW/WHY/WHERE/WHEN (PSUBCAT SFIN)

   17. HOW-TO-INF / 100 (combined with other wh comps)
   (SUBCAT WHVP)
he_PPHS1 explained_VVD how_RGQ to_TO do_VV0 it_PPH1
   (VSUBCAT PP) HOW/WHERE/WHEN (PSUBCAT VPINF)

   18. INF-AC / ??
   ANLT gap (SUBCAT VC_BSE)
he_PPHS1 helped_VVD bake_VV0 the_AT cake_NN1
   (VSUBCAT VPBSE)

   19. ING-NP-OMIT / 242
   (SUBCAT SC_ING, SUBTYPE EQUI)
his_AT hair_NN1 needs_VVZ combing_VVG
   (VSUBCAT VPING)

   20. ING-SC/BE-ING-SC / 21
   (SUBCAT SC_ING, SUBTYPE RAIS)
she_PPHS1 stopped_VVD smoking_VVG
   (VSUBCAT VPING)

   21. ING-AC / ??
   ANLT gap (SUBCAT VC_ING)
she_PPHS1 discussed_VVD writing_VVG novels_NN2
   (VSUBCAT VPING)

   22. INTRANS / 2985
   (SUBCAT NULL)
he_PPHS1 went_VVD
   (VSUBCAT NONE)

   23. INTRANS-RECIP(SUBJ-PL/COORD) / ??
   (SUBCAT NULL)
They_PPHS2 met_VVD
```

```
   *_PP/NN*2 (VSUBCAT NONE)
John_NP1 and_CC her_AT brother_NN1 met_VVD
   *_CC (VSUBCAT NONE) ***

   24. NP / 5281
  (SUBCAT NP) / XTAG:Tnx0Vnx1
he_PPHS1 loved_VVD her_PPHO1
  (VSUBCAT NP)

   25. NP-ADJP / 113
   (SUBCAT OC_AP, SUBTYPE EQUI)
he_PPHS1 painted_VVD the_AT car_NN1 black_JJ
   (VSUBCAT NP_AP)

   26. NP-ADJP-PRED / 46
   (SUBCAT OC_AP, SUBTYPE RAIS)  / XTAG:Tnx0Vs1
she_PPHS1 considered_VVD him_PPHO1 foolish_JJ
   (VSUBCAT NP_AP)

   27. NP-ADVP / 9
   (SUBCAT NP_ADVP)
he_PPHS1 put_VVD it_PPH1 there_RL
   (VSUBCAT NP, PRT +) * there

   28. NP-ADVP-PRED / 281 (with PPs)
   (SUBCAT NP_LOC)   / XTAG:Tnx0Vs1
they_PPHS2 mistakenly_RA thought_VVD him_PPHO1 here_RL
   (VSUBCAT NP, PRT +) here

   29. NP-AS-NP / 3
   (SUBCAT SC_NP_NP, SUBTYPE RAIS, PREP as)
I_PPHS1 sent_VVD him_PPHO1 as_CSA a_AT1 messenger_NN1
   (VSUBCAT NP_PP) (PFORM AS)

   30. NP-AS-NP-SC / 3
   (SUBCAT SC_NP_NP, SUBTYPE RAIS, PREP as)
she_PPHS1 served_VVD the_AT firm_NN1 as_CSA a_AT1 researcher_NN1
   (VSUBCAT NP_PP) (PFORM AS)

   31. NP-FOR-NP / 90
   (SUBCAT NP_PP, SUBTYPE DMOVT, PFORM for)
she_PPHS1 bought_VVD a_AT1 book_NN1 for_IF him_PPHO1
   (VSUBCAT NP_PP) (PFORM FOR)

   32. NP-INF / 11
   (SUBCAT OC_BSE, SUBTYPE RAIS) / XTAG:Tnx0Vs1
he_PPHS1 made_VVD her_PPHO1 sing_VV0
   (VSUBCAT SCOMP)  *_VV0

   33. NP-INF-OC / 17
   (SUBCAT OC_BSE, SUBTYPE EQUI)
he_PPHS1 helped_VVD her_PP$ bake_VV0 the_AT cake_NN1
   (VSUBCAT SCOMP)  *_VV0

   34. NP-ING / 28
   (SUBCAT OC_ING, SUBTYPE RAIS)  / XTAG:Tnx0Vs1
I_PPHS1 kept_VVD them_PPHO2 laughing_VVG
   (VSUBCAT SING)
   35. NP-ING-OC / 45
   (SUBCAT OC_ING, SUBTYPE EQUI)
I_PPHS1 caught_VVD him_PPHO1 stealing_VVG
   (VSUBCAT SING)

   36. NP-ING-SC / ??
   ANLT gap: real complement?
he_PPHS1 combed_VVD the_AT woods_NN2 looking_VVG for_IF her_PPHO1
   (VSUBCAT SING)

   37. NP-NP / 231
   (SUBCAT NP_NP) / XTAG:Tnx0Vnx1nx2
she_PPHS1 asked_VVD him_PPHO1 his_AT name_NN1
   (VSUBCAT NP_NP)

   38. NP-NP-PRED / 38
   (SUBCAT OC_NP, SUBTYPE EQUI)  / XTAG:Tnx0Vs1
they_PPHS2 appointed_VVD him_PPHO1 professor_NN1
   (VSUBCAT NP_NP)

   39. NP-P-ING / 2
   (SUBCAT OC_PP_ING, PFORM from, SUBTYPE PVERB_OR, ORDER POSTNP)
I_PPHS1 prevented_VVD her_PPHO1 from_II leaving_VVG
   (VSUBCAT NP_PP) from (PSUBCAT VPING)

   40. NP-P-ING-OC / 31
   (SUBCAT OC_PP_ING, PFORM, SUBTYPE PVERB_OE, ORDER POSTNP)
I_PPHS1 accused_VVD her_PPHO1 of_IO murdering_VVG her_AT husband_NN1
   (VSUBCAT SING, PRT +) of
   (VSUBCAT NP_PP) * (PSUBCAT VPING)

   41. NP-P-ING-SC / ??
   Gap in ANLT scheme, shld be: (SUBCAT SC_PP_ING, PRT, ORDER POSTNP)
he_PPHS1 wasted_VVD time_NNT1 on_II fussing_VVG with_IW his_AT hair_NN1
   (VSUBCAT NP_PP) on (PSUBCAT VPING)

   42. NP-P-ING-AC / ??
   Gap in ANLT scheme (SUBCAT VC_PP_ING)
he_PPHS1 told_VVD her_PPHO1 about_II climbing_VVG the_AT mountain_NN1
   (VSUBCAT NP_PP) about (PSUBCAT VPING)

   43. NP-P-NP-ING / ??
   ANLT gap (SUBCAT NP_PP_SING)
he_PPHS1 attributed_VVD his_AT failure_NN1 to_II noone_NP1 buying_VVG
his_AT books_NN2
   (VSUBCAT NP_PP) to (PSUBCAT SING)

   44. NP-P-POSSING / ??
   ANLT gap (SUBCAT NP_PP_SING)
They_PPHS2 asked_VVD him_PPHO1 about_II his_PPHO1 participating_VVG
in_II the_AT conference_NN1
   (VSUBCAT NP_PP) about (PSUBCAT SING)

   45. NP-P-WH-S / 0   (not in vppp, and below)
   (SUBCAT NP_WHS, PREP)
they_PPHS2 made_VVD a_AT1 great_JJ fuss_NN1 about_II whether_CSW they_PPHS2
should_VM participate_VV0
```

```
   (VSUBCAT NP_PP) about (PSUBCAT PP) whether (PSUBCAT SFIN)

   46. NP-P-WHAT-S / 0
   (SUBCAT NP_WHS, PREP)
they_PPHS2 made_VVD a_AT1 great_JJ fuss_NN1 about_II what_DDQ they_PPHS2
should_VM do_VV0
   (VSUBCAT NP_WHPP) about (PSUBCAT WHS)

   47. NP-P-WHAT-TO-INF / 0
   (SUBCAT NP_WHVP, PREP)
they_PPHS2 made_VVD a_AT1 great_JJ fuss_NN1 about_II what_DDQ to_TO do_VV0
   (VSUBCAT NP_WHPP) about (PSUBCAT NP)

   48. NP-P-WH-TO-INF / 0
   (SUBCAT NP_WHS, PREP)
they_PPHS2 made_VVD a_AT1 great_JJ fuss_NN1 about_II whether_CSW to_TO go_VV0
   (VSUBCAT NP_PP) about (PSUBCAT PP) whether (PSUBCAT VPINF)

   49. NP-PP / 2010
   (SUBCAT NP_PP, PFORM, SUBTYPE NONE/PVERB?) / XTAG:Tnx0Vnx1pnx2
she_PPHS1 added_VVD the_AT flowers_NN2 to_II the_AT bouquet_NN1
    (VSUBCAT NP_PP) to

   50. NP-PP-PRED / 2010/50??
   (SUBCAT NP_PP, PFORM of, SUBTYPE NONE, PRD +)
I_PPHS1 considered_VVD that_AT problem_NN1 of_IO little_JJ concern_NN1
   (VSUBCAT NP_PPOF)

   51. NP-PRED-RS / 12
   (SUBCAT SC_NP, SUBTYPE RAIS)
he_PPHS1 seemed_VVD a_AT1 fool_NN
   (VSUBCAT NP)

   52. NP-S / 33
   (SUBCAT NP_SFIN, SUBTYPE NONE) / XTAG:Tnx0Vnx1s2
he_PPHS1 told_VVD the_AT audience_NN1 that_CST he_PPHS1 was_VBZ leaving_VVG
   (VSUBCAT NP_SCOMP) * *_VVZ/D/G

   53. NP-TO-INF-OC / 189
   (SUBCAT OC_INF, SUBTYPE EQUI)
I_PPHS1 advised_VVD Mary_NP1 to_TO go_VV0
   (VSUBCAT SINF)

   54. NP-TO-INF-SC / 1
   (SUBCAT SC_NP_INF, SUBTYPE EQUI)
John_NP1 promised_VVD Mary_NP1 to_TO resign_VV0
   (VSUBCAT SINF)

   55. NP-TO-INF-VC / ??
   ANLT gap
they_PPHS2 badgered_VVD him_PPHO1 to_TO go_VV0
    (VSUBCAT SINF)

   56. NP-TO-NP / 105
   (SUBCAT NP_PP, PFORM to, SUBTYPE DMOVT) / XTAG:Tnx0Vnx1Pnx2
he_PPHS1 gave_VVD a_AT1 big_JJ kiss_NN1 to_II his_AT mother_NN1
   (VSUBCAT NP_PP) to
   57. NP-TOBE / 88
    (SUBCAT OC_INF, SUBTYPE RAIS)
I_PPHS1 found_VVD him_PPHO1 to_TO be_VB0 a_AT1 good_JJ doctor_NN1
    (VSUBCAT SINF) BE

   58. NP-VEN-NP-OMIT / 3
   (SUBCAT OC_PASS, SUBTYPE EQUI/RAISING)
he_PPHS1 wanted_VVD the_AT children_NN2 found_VVN
   (VSUBCAT SCOMP) *_VVN

   59. NP-WH-S / 12
   (SUBCAT NP_WHS)
they_PPHS2 asked_VVD him_PPHO1 whether_CSW he_PPHS1 was_VBZ going_VVG
   (VSUBCAT NP_PP) WHETHER/IF (PSUBCAT SFIN)

   60. NP-WHAT-S / 12
   (SUBCAT NP_WHS)
they_PPHS2 asked_VVD him_PPHO1 what_DDQ he_PPHS1 was_VBZ doing_VVG
   (VSUBCAT NP_SCOMP) S(WH +)

   61. NP-WH-TO-INF / 12
   (SUBCAT NP_WHVP)
he_PPHS1 asked_VVD him_PPHO1 whether_CSW to_TO clean_VV0 the_AT house_NN1
   (VSUBCAT NP_PP) WHETHER (PSUBCAT VPINF)

   62. NP-WHAT-TO-INF / 12
   (SUBCAT NP_WHVP)
he_PPHS1 asked_VVD him_PPHO1 what_DDQ to_TO do_VV0
   (VSUBCAT NP_NP) * WHAT/WHO/WHICH

   63. P-ING-SC / 100
   (SUBCAT SC_ING, SUBTYPE EQUI, PREP)
they_PPHS2 failed_VVD in_II attempting_VVG the_AT climb_NN1
   (VSUBCAT PP) in (PSUBCAT VPING)

   64. P-ING-AC / ??
   ANLT gap (SUBCAT VC_ING, PRT)
they_PPHS2 disapproved_VVD of_IO attempting_VVG the_AT climb_NN1
   (VSUBCAT VPING, PRT +) of
they_PPHS2 argued_VVD about_II attempting_VVG the_AT climb_NN1
   (VSUBCAT PP) about (PSUBCAT VPING)

   65. P-NP-ING / 8
   (SUBCAT OC_PP_ING, PFORM @p, SUBTYPE PVERB_OR/OE, ORDER PRENP)
they_PPHS2 worried_VVD about_II him_PPHO1 drinking_VVG
   (VSUBCAT PP) about (PSUBCAT SING)

   66. P-NP-TO-INF(-SC) / 6
   (SUBCAT SC_PP_INF, PFORM @p, SUBTYPE EQUI)
he_PPHS1 conspired_VVD with_IW them_PPHO2 to_TO do_VV0 it_PPH1
   (VSUBCAT PP_VPINF) with (PSUBCAT NP)

   67. P-NP-TO-INF-OC / 29
   (SUBCAT OC_PP_INF, PFORM @p, SUBTYPE PVERB_OE/OR/EQUI)
he_PPHS1 beckoned_VVD to_II him_PPHO1 to_TO come_VV0
      (VSUBCAT PP_VPINF) to (PSUBCAT NP)

   68. P-NP-TO-INF-VC / ??
   ANLT gap
she_PPHS1 appealed_VVD to_II him_PPHO1 to_TO go_VV0
she_PPHS1 appealed_VVD to_II him_PPHO1 to_TO be_VB0 freed_JJ
      (VSUBCAT PP_VPINF) to  (PSUBCAT NP)

   69. P-POSSING / 10
   (SUBCAT OC_PP_ING, PFORM @p, SUBTYPE PVERB_OR, ORDER PRENP)
they_PPHS2 argued_VVD about_II his_PP$ coming_VVG
   (VSUBCAT PP) about (PSUBCAT SING)

   70. P-WH-S / 37
   (SUBCAT WHS, PRT/PREP @p)
he_PPHS1 thought_VVD about_II whether_CSW he_PPHS1 wanted_VVD to_TO go_VV0
   (VSUBCAT PP) about (PSUBCAT PP) WHETHER/IF (PSUBCAT SFIN)

   71. P-WHAT-S / 37
   (SUBCAT WHS, PRT/PREP @p)
he_PPHS1 thought_VVD about_II what_DDQ he_PPHS1 wanted_VVD
   (VSUBCAT WHPP) about (PSUBCAT WHS)

   72. P-WH-TO-INF / 27
   (SUBCAT WHVP, PREP @p)
he_PPHS1 thought_VVD about_II whether_CSW to_TO go_VV0
   (VSUBCAT PP) about (PSUBCAT PP) whether (PSUBCAT VPINF)

   73. P-WHAT-TO-INF / 27
   (SUBCAT WHVP, PREP @p)
he_PPHS1 thought_VVD about_II what_DDQ to_TO do_VV0
   (VSUBCAT WHPP) about

   74. PART / 3219
   (SUBCAT NULL, PRT) / XTAG:Tnx0Vpl
she_PPHS1 gave_VVD up_RL
   (VSUBCAT NONE, PRT +) up
she_PPHS1 gave_VVD up_II
   (VSUBCAT PP) up (PSUBCAT NONE)

   75. PART-ING-SC / 7
   (SUBCAT SC_ING, SUBTYPE EQUI, PRT/PREP)
he_PPHS1 ruled_VVD out_II paying_VVG her_AT debts_NN2
   (VSUBCAT PP) out (PSUBCAT VPING)
he_PPHS1 ruled_VVD out_RP paying_VVG her_AT debts_NN2
   (VSUBCAT VPING, PRT +) out

   76. PART-NP/NP-PART / 2134
  (SUBCAT NP, PRT) (ORDER FREE) / XTAG:Tnx0Vplnx1
I_PPHS1 looked_VVD up_RL the_AT entry_NN1
   (VSUBCAT NP, PRT +) up *
I_PPHS1 looked_VVD the_AT entry_NN1 up_RL
   (VSUBCAT NP, PRT +) * up

   77. PART-NP-PP / 312
   (SUBCAT NP_PP, PFORM, PRT, SUBTYPE NONE/PVERB?)  (ORDER FREE)
I_PPHS1 separated_VVD out_II the_AT three_JJ boys_NN2 from_II the_AT crowd_NN1
   (VSUBCAT PP_PP) out (PSUBCAT NP) from (PSUBCAT NP)
I_PPHS1 separated_VVD out_RL the_AT three_JJ boys_NN2 from_II the_AT crowd_NN1
   (VSUBCAT NP_PP, PRT +) out  from (PSUBCAT NP)
   78. PART-PP / 234
   (SUBCAT PP, PFORM, PRT, SUBTYPE PVERB)
she_PPHS1 looked_VVD in_II on_II her_AT friend_NN1
   (VSUBCAT PP) in (PSUBCAT PP) on (PSUBCAT NP)
she_PPHS1 looked_VVD in_RL on_II her_AT friend_NN1
   (VSUBCAT PP, PRT +) in on (PSUBCAT NP)

   79. PART-WH-S / 20
   (SUBCAT WHS, PRT, SUBTYPE NONE)
they_PPHS2 figured_VVD out_II whether_CSW she_PPHS1 had_VHD n't_XX done_VVD
her_AT job_NN1
   (VSUBCAT PP) out (PSUBCAT PP) WHETHER/IF (PSUBCAT SFIN)
they_PPHS2 figured_VVD out_RP whether_CSW she_PPHS1 had_VHD n't_XX done_VVD
her_AT job_NN1
   (VSUBCAT PP, PRT +) out WHETHER/IF (PSUBCAT SFIN)

   80. PART-WHAT-S / 20
   (SUBCAT WHS, PRT, SUBTYPE NONE)
they_PPHS2 figured_VVD out_II what_DDQ she_PPHS1 had_VHD n't_XX done_VVD
   (VSUBCAT WHPP) out (PSUBCAT WHS)
they_PPHS2 figured_VVD out_RP what_DDQ she_PPHS1 had_VHD n't_XX done_VVD
   (VSUBCAT SCOMP, PRT +) out S(WH +)

   81. PART-WH-TO-INF / 22
   (SUBCAT WHVP, PRT, SUBTYPE NONE)
they_PPHS2 figured_VVD out_II whether_CSW to_TO go_VV0
      (VSUBCAT PP) out (PSUBCAT PP) whether (PSUBCAT VPINF)
they_PPHS2 figured_VVD out_RP whether_CSW to_TO go_VV0
      (VSUBCAT PP, PRT +) out whether (PSUBCAT VPINF)

   82. PART-WHAT-TO-INF / 22
   (SUBCAT WHVP, PRT, SUBTYPE NONE)
they_PPHS2 figured_VVD out_II what_DDQ to_TO do_VV0
   (VSUBCAT WHPP) out (PSUBCAT NP)
they_PPHS2 figured_VVD out_RP what_DDQ to_TO do_VV0
   (VSUBCAT NP, PRT +) WHAT/WHICH/WHO

   83. PART-THAT-S / 48
   (SUBCAT SFIN, PRT, SUBTYPE NONE)
they_PPHS2 figured_VVD out_II that_CST she_PPHS1 had_VHD n't_XX done_VVD
her_AT job_NN1
   (VSUBCAT PP_SCOMP) out (PSUBCAT NONE) *_VVG/Z/D
they_PPHS2 figured_VVD out_RP that_CST she_PPHS1 had_VHD n't_XX done_VVD
her_AT job_NN1
   (VSUBCAT SCOMP, PRT +) out *_VVG/Z/D

   84. POSSING / 27
   (SUBCAT OC_ING, SUBTYPE RAIS)
he_PPHS1 dismissed_VVD their_PP$ writing_VVG novels_NN2
   (VSUBCAT SING)

   85. POSSING-PP / ??
   ANLT gap (SUBCAT OC_ING_PP)
she_PPHS1 attributed_VVD his_PP$ drinking_VVG too_RA much_RA to_II his_AT anxiety_NN1
   (VSUBCAT SING_PP) to (PSUBCAT NP)

   86. ING-PP / ??
   ANLT gap
```

```
they_PPHS2 limited_VVD smoking_VVG a_AT pipe_NN1 to_II the_AT lounge_NN1
   (VSUBCAT VPING_PP) to (PSUBCAT NP)

   87. PP / 2465 (366 LOC)
   (SUBCAT PP/LOC, PFORM, SUBTYPE NONE/PVERB) / XTAG:Tnx0Vpnx1
they_PPHS2 apologized_VVD to_II him_PPHO1
   (VSUBCAT PP) to (PSUBCAT NP)

   88. PP-FOR-TO-INF / 1
   (SUBCAT PP_SINF, PFORM)
they_PPHS2 contracted_VVD with_IW him_PPHO1 for_IF the_AT man_NN1 to_TO go_VV0
   (VSUBCAT PP_PP) with (PSUBCAT NP) for (PSUBCAT SINF)

   89. PP-HOW-S / 7
   (SUBCAT PP_WHS, PFORM)
he_PPHS1 explained_VVD to_II her_PPHO1 how_RGQ she_PPHS1 did_VDD it_PPH1
   (VSUBCAT PP_PP) to (PSUBCAT NP) HOW/WHY/WHERE/WHEN (PSUBCAT SFIN)

   90. PP-HOW-TO-INF / 3
   (SUBCAT PP_WHVP, PFORM)
he_PPHS1 explained_VVD to_II them_PPHO2 how_RGQ to_TO do_VV0 it_PPH1
   (VSUBCAT PP_PP) to (PSUBCAT NP) HOW/WHERE/WHEN (PSUBCAT VPINF)

   91. PP-P-WH-S / ??
   Gap in ANLT scheme: (SUBCAT PP_WHS, PFORM, PRT)
I_PPHS1 agreed_VVD with_IW him_PPHO1 about_II whether_CSW he_PPHS1 should_VM
kill_VV0 the_AT peasants_NN2
   (VSUBCAT PP_PP) with (PSUBCAT NP) about (PSUBCAT PP) WHETHER (PSUBCAT SFIN)

   92. PP-P-WHAT-S / ??
   Gap in ANLT scheme
I_PPHS1 agreed_VVD with_IW him_PPHO1 about_II what_DDQ he_PPHS1 should_VM do_VV0
   (VSUBCAT PP_WHPP) with (PSUBCAT NP) about (PSUBCAT WHS)

   93. PP-P-WHAT-TO-INF / ??
   Gap in ANLT scheme
I_PPHS1 agreed_VVD with_IW him_PPHO1 about_II what_DDQ to_TO do_VV0
   (VSUBCAT PP_WHPP) with (PSUBCAT NP) about (PSUBCAT NP)

   94. PP-P-WH-TO-INF / ??
   Gap in ANLT scheme
I_PPHS1 agreed_VVD with_IW him_PPHO1 about_II whether_CSW to_TO go_VV0
   (VSUBCAT PP_PP) with (PSUBCAT NP) about (PSUBCAT PP) whether (PSUBCAT VPINF)

   95. PP-PP / 64 (22 PVERB)
   (SUBCAT PP_PP)
they_PPHS2 flew_VVD from_II London_NP1 to_II Rome_NP1
   (VSUBCAT PP_PP) from (PSUBCAT NP) to (PSUBCAT NP)

   96. PP-PRED-RS / 0 (not in vppp)
   (SUBCAT PP, SUBTYPE RAIS)
the_AT matter_NN1 seems_VVZ in_II dispute_NN1
   (VSUBCAT PP) in (PSUBCAT NP)

   97. PP-THAT-S / 22
   (SUBCAT PP_SFIN, SUBTYPE NONE, PFORM)
they_PPHS2 admitted_VVD to_II the_AT authorities_NN2 that_CST they_PPHS2
had_VHD entered_VVD illegally_RA
```

```
   (VSUBCAT PP_SCOMP) to (PSUBCAT NP)  *_VVD/Z/G

   98. PP-THAT-S-SUBJUNCT / 2
   (SUBCAT PP_SBSE, PFORM, S[BSE, that])
they_PPHS2 suggested_VVD to_II him_PPHO1 that_CST he_PPHS1 go_VV0
   (VSUBCAT PP_SCOMP) to (PSUBCAT NP)  *_VV0

   99. PP-TO-INF-RS / 1
   (SUBCAT SC_PP_INF, SUBTYPE RAIS, PFORM, VP[to])
he_PPHS1 appeared_VVD to_II her_PPHO1 to_TO be_VB0 ill_JJ
   (VSUBCAT PP_VPINF) to (PSUBCAT NP) BE

   100. PP-WH-S / 7
   (SUBCAT PP_WHS, PFORM)
they_PPHS2 asked_VVD about_II everybody_NP1 whether_CSW they_PPHS2
had_VHD enrolled_VVN
   (VSUBCAT PP_PP) about (PSUBCAT NP) WHETHER/IF (PSUBCAT SFIN)

   101. PP-WHAT-S / 7
   (SUBCAT PP_WHS, PFORM)
they_PPHS2 asked_VVD about_II everybody_NP1 what_DDQ they_PPHS2 had_VHD done_VVN
   (VSUBCAT PP_WHS) about (PSUBCAT NP)

   102. PP-WH-TO_INF / 3
   (SUBCAT PP_WHVP)
they_PPHS2 deduced_VVD from_II kim_NP1 whether_CSW to_TO go_VV0
   (VSUBCAT PP_PP) from (PSUBCAT NP) whether (PSUBCAT VPINF)

   103. PP-WHAT-TO-INF / 3
   (SUBCAT PP_WHVP)
they_PPHS2 deduced_VVD from_II kim_NP1 what_DDQ to_TO do_VV0
   (VSUBCAT PP_WHVP) from (PSUBCAT NP) WHAT/WHO/WHICH

   104. S / 296
   (SUBCAT SFIN, SUBTYPE NONE) / XTAG:Tnx0Vs1
they_PPHS2 thought_VVD that_CST he_PPHS1 was_VBZ always_RA late_JJ
   (VSUBCAT SCOMP) *_VVD/Z/G

   105. S-SUBJ-S-OBJ / 9
   (SUBCAT SFIN, SUBTYPE EXTRAP, AGR S[FIN -])
for_IF him_PPHO1 to_TO report_VV0 the_AT theft_NN1 indicates_VVD that_CST
he_PPHS1 was_VBZ n't_XX guilty_JJ
   *_VV0 (VSUBCAT SCOMP)  *_VVD/Z/G

   106. S-SUBJUNCT / 27
   (SUBCAT SBSE)
She_PPHS1 demanded_VVD that_CST he_PPHS1 leave_VV0 immediately_RA
   (VSUBCAT SCOMP) *_VV0

   107. SEEM-S / 9
   (SUBCAT SFIN, SUBTYPE EXTRAP, AGR N2[NFORM IT])
it_PPH1 seems_VVZ that_CST they_PPHS2 left_VVD
   IT (VSUBCAT SCOMP) *_VVD/Z/G

   108. SEEM-TO-NP-S / 1
   (SUBCAT PP_SFIN, SUBTYPE EXTRAP, PFORM, AGR N2[NFORM IT])
it_PPH1 seems_VVZ to_II her_PPHO1 that_CST they_PPHS2 were_VBDR wrong_JJ
   IT (VSUBCAT PP_SCOMP) to (PSUBCAT NP) *_VVD/Z/G
   109. THAT-S / 296 (with 104)
   (SUBCAT SFIN, SUBTYPE NONE) / XTAG:Tnx0Vs1
he_PPHS1 complained_VVD that_CST they_PPHS2 were_VBDR coming_VVG
   (VSUBCAT SCOMP) *_VVD/Z/G

   110. TO-INF-AC  / ??
   ANLT gap (SUBCAT VC_INF)
He_PPHS1 helped_VVD to_TO save_VV0 the_AT child_NN1
   (VSUBCAT VPINF)

   111. TO-INF-RS / 27
   (SUBCAT SC_INF, SUBTYPE RAIS)
he_PPHS1 seemed_VVD to_TO come_VV0
   (VSUBCAT VPINF) be

   112. TO-INF-SC / 179
  (SUBCAT SC_INF, SUBTYPE EQUI)
I_PPHS1 wanted_VVD to_TO come_VV0
   (VSUBCAT VPINF)

   113.WH-S / 133
   (SUBCAT WHS) / XTAG:Tnx0Vs1
he_PPHS1 asked_VVD whether_CSW he_PPHS1 should_VM come_VV0
   (VSUBCAT PP) WHETHER/IF (PSUBCAT SFIN)

   114. WHAT-S / 133
  (SUBCAT WHS) / XTAG:Tnx0Vs1
he_PPHS1 asked_VVD what_DDQ he_PPHS1 should_VM do_VV0
  (VSUBCAT SCOMP) S(WH +)

   115. WH-TO-INF / 78
  (SUBCAT WHVP) / XTAG:Tnx0Vs1
he_PPHS1 asked_VVD whether_CSW to_TO clean_VV0 the_AT house_NN1
  (VSUBCAT PP) whether (PSUBCAT VPINF)

   116. WHAT-TO-INF / 78
  (SUBCAT WHVP) / XTAG:Tnx0Vs1
he_PPHS1 asked_VVD what_DDQ to_TO do_VV0
  (VSUBCAT NP) WHAT/WHO/WHICH

  117. XTAG:Tnx0Vplnx1nx2 / 45
  (SUBCAT NP_NP, PRT)
I_PPHS1 opened_VVD him_PPHO1 up_RP a_AT new_JJ bank_NN1 account_NN1
  (VSUBCAT NP_NP, PRT +) up

  118. XTAG:Light-verbs (various classes) /  ??
  ANLT gaps (not a genuine class as subclasses of 49/50)
he_PPHS1 made_VVD comments_NN2 on_II the_AT paper_NN1
  (VSUBCAT NP_PP) (make comments) on (PSUBCAT NP)

  119. (SUBCAT PP/LOC, PFORM, PRT, SUBTYPE NONE) / 881 (LOC 45)
he_PPHS1 breaks_VVZ away_RP from_II the_AT abbey_NN1
  (VSUBCAT PP, PRT +) away from (PSUBCAT NP)

  120. (SUBCAT NP_PP, PFORM, PRT, SUBTYPE DMOVT) / 25
he_PPHS1 brought_VVD a_AT book_NN1 back_RP for_IF me_PPHO1
  (VSUBCAT NP_PP, PRT +) back for (PSUBCAT NP)
  121. (SUBCAT PP_PP, PFORM, PRT) / 3
he_PPHS1 came_VVD down_RP on_II him_PPHO1 for_IF his_AT bad_JJ behaviour_NN1
  (VSUBCAT PP_PP, PRT +) down on (PSUBCAT NP) for (PSUBCAT NP)

  122. (SUBCAT NP_PP_PP, PFORM) / 16
he_PPHS1 turned_VVD it_PPHO1 from_II a_AT disaster_NN1 into_II a_AT victory_NN1
  (VSUBCAT NP_PP_PP) from (PSUBCAT NP) into (PSUBCAT NP)

  123. (SUBCAT MP) / 29
it_PPHS1 cost_VVD ten_MC pounds_NNU2
  (VSUBCAT NP) _NNU/(NTYPE MEAS)

  124. (SUBCAT NP_MP) / 6
it_PPHS1 cost_VVD him_PPHO1 ten_MC pounds_NNU2
  (VSUBCAT NP_NP) _NNU/(NTYPE MEAS)

  125. (SUBCAT NP_MP, PRT) / 1
it_PPHS1 set_VVD him_PPHO1 back_RP ten_MC pounds_NNU2
  (VSUBCAT NP_NP, PRT +) back _NNU/(NTYPE MEAS)

  126. (SUBCAT ADL, PRT) / 13
he_PPHS1 came_VVD off_RP badly_RP
  (VSUBCAT NONE, PRT +) off (...PRT +) badly

  127. (SUBCAT ADV_PP, PFORM) / 2
things_NN2 augur_VV0 well_RP for_IF him_PPHO1
  (VSUBCAT PP, PRT +) well for (PSUBCAT NP)

  128. (SUBCAT SFIN, AGR N2[NFORM IT], PRT) / 3
it_PPHS1 turns_VVZ out_RP that_CST he_PPHS1 did_VVD it_PPHO1
  IT (VSUBCAT SCOMP, PRT +) out *_VVD/Z/G

  129. (SUBCAT SFIN, AGR S[FIN +], SUBTYPE EXTRAP) / 9
that_CST he_PPHS1 came_VVD matters_VVZ
  *_VVD/G/Z (VSUBCAT NONE)

  130. (SUBCAT NP_SFIN, SUBTYPE NONE, PRT) / 4
he_PPHS1 had_VVD her_PPHO1 on_RP that_CST he_PPHO1 attended_VVD
  (VSUBCAT NP_SCOMP, PRT +) on *_VVD/Z/G

  131. (SUBCAT PP_SFIN, SUBTYPE NONE, PRT) / 4
she_PPHS1 gets_VVZ through_RP to_II him_PPHO1 that_CST he_PPHS1 came_VVD
  (VSUBCAT PP_SCOMP, PRT +) through to (PSUBCAT NP) *_VVD/Z/G

  132. (SUBCAT NP_NP_SFIN) / 4
he_PPHS1 bet_VVD her_PPHO1 ten_MC pounds_NNU2 that_CST he_PPHS1 came_VVD
  (VSUBCAT NP_NP_SCOMP) _NNU*/(NTYPE MEAS) *_VVD/Z/G

  133. (SUBCAT NP_SBSE) / 1
he_PPHS1 petitioned_VVD them_PPHO2 that_CST he_PPHS1 be_VB0 freed_VVN
  (VSUBCAT NP_SCOMP) * *_VB0

  134. (SUBCAT IT_WHS, SUBTYPE IF, AGR N2[NFORM IT]) / 1
I_PPHS1 would_VM appreciate_VV0 it_PPHO1 if_CF he_PPHS1 came_VVD
  (VSUBCAT NP_PP) if (PSUBCAT SFIN)

  135. (SUBCAT PP_WHS, PFORM, AGR N2[NFORM IT]) / 1
```

```
it_PPHS1 dawned_VVD on_II him_PPHO1 what_DDQ he_PPHS1 should_VM do_VV0
  IT (VSUBCAT PP_WHS) on (PSUBCAT NP)

  136. (SUBCAT SC_NP, PRT, SUBTYPE RAIS/EQUI, PRD +) / 2
he_PPHS1 turned_VVD out_RP a_AT fool_NN1
  (VSUBCAT NP, PRT +) out

  137. (SUBCAT SC_AP, PRT, SUBTYPE EQUI/RAIS) / 22 (RAIS 3)
he_PPHS1 started_VVD out_RP poor_JJ
  (VSUBCAT AP, PRT +) out
he_PPHS1 started_VVD out_II poor_JJ
  (VSUBCAT PP_AP) out (PSUBCAT NONE)

  138. (SUBCAT SC_INF, PRT, SUBTYPE RAIS) / 6
he_PPHS1 turned_VVD out_RP to_TO be_VB0 a_AT crook_NN1
  (VSUBCAT VPINF, PRT +) out BE
he_PPHS1 turned_VVD out_II to_TO be_VB0 a_AT crook_NN1
  (VSUBCAT PP_VPINF) out (PSUBCAT NONE) BE

  139. (SUBCAT SC_INF, PRT, SUBTYPE EQUI) / 12
he_PPHS1 set_VVD out_RP to_TO win_VV0
  (VSUBCAT VPINF, PRT +) out
he_PPHS1 set_VVD out_II to_TO win_VV0
  (VSUBCAT PP_VPINF) out (PSUBCAT NONE)

  140. (SUBCAT SC_ING, PREP, PRT, SUBTYPE EQUI) / 32
he_PPHS1 got_VVD around_RP to_II leaving_VVG
  (VSUBCAT PP, PRT +) around to (PSUBCAT VPING)

  141. (SUBCAT SC_PASS, SUBTYPE RAIS) / 4
he_PPHS1 got_VVD given_VVN a_AT book_NN1
  (VSUBCAT VPPRT)

  142. (SUBCAT SC_BSE, SUBTYPE EQUI) / 3
he_PPHS1 dared_VVD dance_VV0
  (VSUBCAT VPBSE)

  143. (SUBCAT SC_NP_AP, SUBTYPE RAIS, PREP as) / 3
he_PPHS1 strikes_VVZ me_PPHO1 as_CSA foolish_JJ
  (VSUBCAT NP_PP) AS (PSUBCAT AP)

  144. (SUBCAT OC_NP, SUBTYPE RAIS) / 35
he_PPHS1 considers_VVZ Fido_NP1 a_AT fool_NN1
  (VSUBCAT NP_NP)

  145. (SUBCAT OC_AP, SUBTYPE RAIS, PRT) / 3
he_PPHS1 makes_VVD him_PPHO1 out_RP crazy_JJ
  (VSUBCAT NP_AP, PRT +) out

  146. (SUBCAT OC_AP, SUBTYPE EQUI, PRT) / 4
he_PPHS1 sands_VVZ it_PPHO1 down_RP smooth_JJ
  (VSUBCAT NP_AP, PRT +) down

  147. (SUBCAT OC_AP, SUBTYPE EQUI, PREP as) / 5
he_PPHS1 condemned_VVD him_PPHO1 as_CSA stupid_JJ
  (VSUBCAT NP_PP) AS (PSUBCAT AP)

  148. (SUBCAT OC_AP, SUBTYPE EQUI, PREP as, PRT) / 6
```

```
he_PPHS1 put_VVD him_PPHO1 down_RP as_CSA stupid_JJ
  (VSUBCAT NP_PP, PRT +) down AS (PSUBCAT AP)

  149. (SUBCAT OC_INF, SUBTYPE RAIS, PRT) / 3
he_PPHS1 made_VVD him_PPHO1 out_RP to_TO be_VV0 crazy_JJ
  (VSUBCAT SINF, PRT +) out BE

  150. (SUBCAT OC_INF, SUBTYPE EQUI, PRT) / 19
he_PPHS1 spurred_VVD him_PPHO1 on_RP to_TO try_VV0
  (VSUBCAT SINF, PRT +) on

  151. (SUBCAT OC_PP_INF, SUBTYPE PVERB_OE, PFORM, PRT) / 6
he_PPHS1 kept_VVD on_RP at_II him_PPHO1 to_TO join_VV0
  (VSUBCAT PP_VPINF, PRT +) on at (PSUBCAT NP)

  152. (SUBCAT OC_PP_ING, SUBTYPE PVERB_OE, PFORM, PRT) / 4
he_PPHS1 talked_VVD him_PPHO1 around_RP into_II leaving_VVG
  (VSUBCAT NP_PP, PRT +) around into (PSUBCAT VPING)

  153. (SUBCAT OC_PP_BSE, PFORM, SUBTYPE PVERB_OE) / 1
he_PPHS1 looked_VVD at_II him_PPHO1 leave_VV0
  (VSUBCAT PP_SCOMP) at (PSUBCAT NONE) *_VV0

  154. (SUBCAT VPINF, SUBTYPE EXTRAP, AGR VP[FIN-]) / 4
to_TO see_VV0 them_PPHO2 hurts_VVZ
  _VV0 (VSUBCAT NONE)

  155. (SUBCAT NP_ADL) / 39
he_PPHS1 stood_VVD it_PPHO1 alone_RL
  (VSUBCAT NP, PRT +) * *_RL/A/P

  156. *NP-HOW-S / ?
he_PPHS1 asked_VVD him_PPHO1 how_RGQ he_PPHS1 came_VVD
  (VSUBCAT NP_PP) HOW/WHY/WHERE/WHEN (PSUBCAT SFIN)

  157. *NP-FOR-TO-INF / ?
he_PPHS1 gave_VVD money_NN2 for_IF him_PPHO1 to_TO go_VV0
  (VSUBCAT NP_PP FOR (PSUBCAT SINF)

  158. *IT-PASS-SFIN / ?
it_PPHS1 is_VBZ believed_VVN that_CST he_PPHS1 came_VVD
  IT PASS (VSUBCAT SCOMP)

  159. *AS-IF-SFIN / ?
he_PPHS1 seems_VVZ as_CS if_CS he_PPHS1 is_VBZ clever_JJ
  (VSUBCAT PP) AS (PSUBCAT PP) IF (PSUBCAT SFIN)

  160. (SUBCAT ADL)
it_PPHS1 carves_VVZ easily_RP
  (VSUBCAT NONE) *_RP/A

  161. (SUBCAT SC_NP SUBTYPE EQUI)
he_PPHS1 felt_VVD a_AT fool_NN1
  (VSUBCAT NP)

  162. *AS-VPPRT
he_PPHS1 accepted_VVD him_PPHO1 as_II/CSA associated_VVN
  (VSUBCAT NP_PP) AS  (PSUBCAT VPPRT)
  163. *AS-VPING
he_PPHS1 accepted_VVD him_PPHO1 as_II/CSA being_VBG normal_JJ
  (VSUBCAT NP_PP) AS  (PSUBCAT VPING)
```

Appendix B

Test Verbs from Chapter 6

Tables B.1 and B.2 list the 334 unclassified test verbs used for experiments reported in section 6.4.

accept	*comfort*	*endure*	*infer*	*prevail*	*stay*
accommodate	*compensate*	*enforce*	*inform*	*prevent*	*steer*
account	*complement*	*engage*	*insist*	*price*	*stop*
accuse	***complete***	*ensure*	*insure*	*print*	*strengthen*
achieve	*compose*	*equate*	*intend*	***produce***	*stretch*
acknowledge	*compress*	***establish***	*introduce*	*promise*	*strive*
act	*concern*	*exemplify*	*invade*	*propose*	*study*
adjust	*conduct*	*exercise*	*involve*	*prove*	*subject*
admit	*confess*	*exert*	*issue*	*quit*	*succeed*
advise	*confine*	*exist*	*jar*	*raise*	*sue*
affect	*conform*	*expand*	*justify*	***react***	*suffer*
afflict	*confront*	***expect***	*know*	*read*	*suggest*
afford	***consider***	*expire*	*lack*	*recognize*	*suit*
aid	*consist*	*expose*	*launch*	*recommend*	*support*
allow	*constitute*	*express*	*learn*	*reconvene*	*swear*
amend	*contain*	*explain*	*leer*	*reduce*	*take*
announce	*contend*	*extend*	*let*	*refer*	*talk*
answer	*contest*	*face*	*lighten*	*register*	*tangle*
appear	*continue*	*fail*	***like***	*regroup*	*taste*
apply	*control*	*fascinate*	*limit*	*remain*	*teach*
approach	*converse*	*fear*	*list*	*repair*	*tell*

Table B.1: Unclassified test verbs I

approve	*cool*	*feed*	*live*	*repeat*	*tempt*
arise	*cope*	***feel***	*look*	*replace*	*tend*
ask	*counsel*	*figure*	*lose*	*reply*	*term*
aspire	*crack*	***find***	*love*	*report*	***terminate***
assail	*create*	*fit*	*make*	*represent*	*test*
astonish	*deal*	***fix***	*match*	*require*	*thank*
attempt	*deceive*	*flow*	*measure*	*rescent*	*thicken*
attend	***decide***	*force*	*meditate*	*resemble*	*think*
back	*declare*	*forsake*	***meet***	*resign*	*threaten*
become	*decry*	*freeze*	*melt*	*resist*	*time*
begin	*depend*	*glance*	*metamorphose*	*resolve*	*toy*
believe	*derive*	*greet*	*mirror*	*respond*	*transcend*
bend	*design*	*grow*	*miss*	*restrain*	*transpire*
bet	*desire*	*guarantee*	*mock*	*retire*	*treat*
boom	*despair*	*guard*	*motivate*	*revere*	*trigger*
border	*detail*	*hail*	*note*	*review*	*tripple*
bother	*deteriorate*	*handle*	*nourish*	*rise*	*trust*
break	*determine*	***happen***	*object*	*rule*	*try*
brood	*develop*	*hate*	*observe*	*say*	*understand*
build	*devise*	*hear*	*occupy*	*schedule*	*understate*
bunch	*dictate*	***help***	*open*	*scream*	*urge*
call	*die*	*hold*	*oppose*	*see*	*use*
campaign	*disarm*	*honour*	*participate*	***seem***	*view*
capture	*discover*	*hope*	*peep*	*serve*	*violate*
care	*discuss*	*hurt*	*penalize*	*share*	*voice*
cause	*disdain*	*idolize*	*permit*	*shatter*	*vote*
celebrate	*disorder*	***ignore***	*persist*	*shout*	*wait*
centralize	*display*	*illuminate*	*pipe*	*show*	*want*
challenge	*dress*	*illustrate*	*plan*	*sin*	*watch*
change	*drift*	*impair*	*ponder*	*sniff*	*wonder*
characterize	*dwindle*	*imply*	*postulate*	***solve***	*work*
choose	*elaborate*	*inaugurate*	*predict*	*speak*	*worship*
cite	*elect*	*include*	*prepare*	*speculate*	*write*
claim	***emerge***	*increase*	*preserve*	***start***	
clothe	***end***	*induce*	*preside*	*state*	

Table B.2: Unclassified test verbs II

Appendix C

Diathesis Alternation - SCF Mapping

In Chapter 7 (section 7.2.3), we briefly discussed McCarthy's work on diathesis alternation acquisition (McCarthy and Korhonen, 1998; McCarthy, 2001). We contributed to this work by producing a mapping between SCFs involved in Levin alternations (Levin, 1993) and those recognized by Briscoe and Carroll's system (Appendix A). This source was employed by McCarthy for selecting candidate SCFs and verbs for alternations.

In constructing this mapping, each Levin alternation was first assigned a shallow syntactic description, based on example sentences given in Levin (1993). All SCFs matching this syntactic description were then extracted from the list of 163 SCFs (see Appendix A). The outcome was checked manually for final SCF assignments. The resulting set of SCFs provides in most cases more detailed syntactic description of an alternation than that provided by Levin. Levin's example sentences often exemplify only the most prototypical frames involved in an alternation. In reality, many alternations can occur with a wider range of frames.

Where possible, we supplemented the syntactic description of an alternation with constraints or preferences on argument slots, possible prepositions and participating verbs. We based these constraints/preferences on information provided in Levin (1993). Preferences on argument slots were defined as simple descriptive labels and WordNet conceptual classes. The latter were identified manually from the noun hierarchy of the taxonomy[1]. Allowable prepositions were simply given as a list of lemmas. Participating verbs were defined as Levin verb classes involved in an alternation. The resulting constraints/preferences are often vague, either because the description given by Levin is inadequate or simply because, in some cases, no strong constraints exist, due to the semi-productive and elusive nature of alternations.

Figure C.1 displays a sample entry from the alternation-SCF mapping for the instrument subject alternation. It shows firstly a pair of example sentences from Levin (1993) where the alternation occurs and below it, a simple syntactic description for the alternation. This is followed by description of preferences/constraints. These

[1] We used for this work WordNet version 1.5.

3.3 Instrument Subject Alternation	
Example	*David broke the window with the hammer* ⇔ *The hammer broke the window*
Syntax	NP1 V NP2 P NP3 ⇔ NP3 V NP2
Constraints	**NP1**: (in)animate entity, WN class 100002403 **NP2**: breakable physical object, WN class 100009469 **NP3**: intermediary instrument, WN class 102009476 **P**: *with* **V**: *Break* verbs
Alternating slot(s)	NP2, NP3
Alternating SCFs	49 ⇔ 24, 77 ⇔ 76
Further description	Only *Break* verbs that take intermediary instruments may participate. These are change of state verbs which refer to actions that bring about a change in the 'material integrity' of some entity. Their meaning provides no information on how the change of state came about. Example verbs: *break, chip, crash, crush, fracture, rip*

Figure C.1: A sample entry for instrument subject alternation

Alternation Category	**Example Alternation**
Extraposition	*To read pleases them* ⇔ *It pleases them to read* SCF 8 ⇔ SCF 11
Equi	*I advised Mary to go* ⇔ *I advised Mary* SCF 53 ⇔ SCF 24
Raising	*Julie strikes me as foolish* ⇔ *Julie strikes me as a fool* SCF 143 ⇔ SCF 29
Category switch	*He failed in attempting to climb* ⇔ *He failed in the climb* SCF 63 ⇔ SCF 87
PP deletion	*Phil explained to him how to do it* ⇔ *Phil explained how to do it* SCF 90 ⇔ SCF 17
P deletion	*I prefer for her to do it* ⇔ *I prefer her to do it* SCF 15 ⇔ SCF 53

Table C.1: Examples of new alternations

indicate that the alternation typically applies to Levin "*Break*" verbs permitting the preposition *with* and taking three noun phrases capable of functioning as (in)animate entity, breakable physical object and intermediary instrument, respectively. After this, the slots and SCFs involved in the alternation are specified. The latter are given as SCF numbers recognized by Briscoe and Carroll's system. Frames 49 and 24 alternate in the example given by Levin. Frames 77 and 76 alternate in another, phrasal/prepositional verb variant, not exemplified in Levin (1993), e.g. *David broke the door down with the axe* ⇔ *The axe broke the door down.* Finally, some further details of the alternation are given.

Levin's classification covers mostly alternations involving NP and PP complements. Those involving control or sentential complements are largely ignored. Although individual studies are available on a few alternations or verb classes taking sentential complements (e.g. Alexander and Kunz, 1965; Rudanko, 1989; Jackendoff, 1990), no extensive Levin style reference work exists which would cover them. After completing the Levin-SCF part of the mapping, we screened through the list of 163 SCFs, considering possible further alternations between pairs of SCFs, especially those involving control and sentential complements. We used criteria similar to Levin's for recognition of alternations: the SCFs alternating should preserve the sense in question, or modify it systematically.

Several additional alternations were discovered and grouped into different categories: alternations involving exraposition, equi, raising, category switch, PP deletion and P deletion. Table C.1 shows an example alternation from each category. Further work is required on these alternations before we can group them into semantically motivated verb classes.

Bibliography

Abney, S. 1991. Parsing by chunks. In Berwick, R., Abney, S., and Tenny S. eds. *Principle-Based Parsing.* Kluwer Academic Publishers, Dordrecht: 257–278.

Abney, S. 1996. Partial parsing via finite-state cascades. *Natural Language Engineering* 2(4): 337–344.

Alexander, D. and Kunz, W. J. 1964. *Some Classes of Verbs in English.* University Linguistics Club, Bloomington.

Baum, L. E. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov Processes. *Inequalities* 3(1): 1–8.

Bloomfield, L. 1933. *Language.* Allen & Unwin, London.

Boguraev, B. and Briscoe, E. J. 1987. Large lexicons for natural language processing: utilising the grammar coding system of the Longman Dictionary of Contemporary English. *Computational Linguistics* 13.4: 219–240.

Boguraev, B. and Briscoe, E. J. (eds.) 1989. *Computational Lexicography for Natural Language Processing.* Longman, London.

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

Boguraev, B., Briscoe, E. J., and Copestake, A. 1991. *Database Models for Computational Lexicography.* Research Report RC 17120, IBM Research Center, Yorktown Heights, New York.

Boguraev, B. and Pustojevsky, J. (eds.) 1995. *Corpus Processing for Lexical Acquisition.* MIT Press, Cambridge, Massachusetts.

Brent, M. 1991. Automatic acquisition of subcategorization frames from untagged text. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, CA. 209–214.

Brent, M. 1993. From grammar to lexicon: unsupervised learning of lexical syntax. *Computational Linguistics* 19.3: 243–262.

Bresnan, J. 1976. On the form and functioning of transformations. *Linguistic Inquiry* 7: 3–40.

Bresnan, J. (ed.) 1982. *The Mental Representation of Grammatical Relations.* MIT Press, Cambridge, Massachusetts.

Bresnan, J. and Kanerva, J. 1989. Locative inversion in Chichewa: a case study of factorization in grammar. *Linguistic Inquiry* 21: 1–50.

Briscoe, E. J. 1991. Lexical issues in natural language processing. In Klein, E. and Veltman, F. eds. *Natural Language and Speech.* Springer-Verlag, Berlin, Heidelberg, New York: 39–68.

Briscoe, E. J. 2000. *Dictionary and System Subcategorisation Code Mappings.* Unpublished manuscript, University of Cambridge Computer Laboratory.

Briscoe, E. J. 2001. From dictionary to corpus to self-organizing dictionary: learning valency associations in the face of variation and change. In *Proceedings of the Corpus Linguistics*, Lancaster University, UK.

Briscoe, E. J. and Carroll, J. 1993. Generalized probabilistic LR parsing for unification-based grammars. *Computational Linguistics* 19.1: 25–60.

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

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

Briscoe, E. J. and Carroll, J. 2000. *Grammatical Relation Annotation.* http://www.cogs.susx.ac.uk/lab/nlp/carroll/grdescription/index.html.

Briscoe, E. J., Carroll, J., and Korhonen, A. 1997. *Automatic extraction of subcategorization frames from corpora - a framework and 3 experiments.* Sparkle WP5 Deliverable. http://www.ilc.pi.cnr.it/.

Briscoe, E. J. and Copestake, A. 1999. Lexical rules in constraint-based grammars. *Computational Linguistics* 25.4: 487–526.

Briscoe, E. J., de Paiva, V., and Copestake, A. 1993. *Inheritance, Defaults and the Lexicon.* Cambridge University Press, Cambridge.

Byrd, R. J., Calzolari, N., Chodorow, M. S., Klavans, J. L., Neff, M. S., and Rizk, O.

A. 1987. Tools and methods for computational lexicology. *Computational Linguistics* 13(3-4): 219–240.

Cambridge University Press, Editor. 1995. *Cambridge International Dictionary of English.* Cambridge University Press, Cambridge.

Carpenter, R. 1992. *The logic of typed feature structures.* Cambridge University Press, Cambridge.

Carpenter, R. 1993. Sceptical and credulous default unification with application to templates and inheritance. In Briscoe, E. J., Copestake, A., and de Paiva, V. eds. *Inheritance, Defaults and the Lexicon.* Cambridge University Press, Cambridge: 13–37.

Carroll, J. 1993. *Practical unification-based parsing of natural language.* Cambridge University Computer Laboratory, TR-224.

Carroll, J. 1994. Relating complexity to practical performance in parsing with wide-coverage unification grammars. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, NMSU, Las Cruces, NM. 287–294.

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

Carroll, J., Briscoe, E. J., Calzolari, N., Federici, S., Montemagni, S., Pirrelli, V., Grefenstette, G., Sanfilippo, A., Carroll, G., and Rooth, M. 1997. *Specification of Phrasal Parsing.* Sparkle WP1 Deliverable. http://www.ilc.pi.cnr.it/.

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

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

Carroll, G. and Rooth, M. 1998. Valence induction with a head-lexicalized PCFG. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing*, Granada, Spain. 36–45.

Carter, D. M. 1989. *Lexical Acquisition in the Core Language Engine.* Report CRC-019. http://www.cam.sri.com/tr/.

Casella, G. and Berger, R.L. 1990. *Statistical Inference.* Wadsworth, California.

Casti, J. L. 1994. *Complexification.* Harper Collins, New York.

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

Chitrao, M. and Grishman, R. 1990. Statistical parsing of messages. In *Proceedings of the Darpa Speech and Natural Language Workshop*, Hidden Valley, PA. 263–266.

Chomsky, N. 1965. *Aspects of the Theory of Syntax.* MIT Press, Cambridge, Massachusetts.

Chomsky, N. 1970. Remarks on Nominalization. In Jacobs, R. and Rosenbaum, P. eds. *Readings in English Transformational Grammar.* Ginn, Waltham, Massachusetts: 184–221.

Chomsky, N. 1981. *Lectures on Government and Binding.* Dordrecht, Foris.

Christ, O. 1994. *The IMS Corpus Workbench Technical Manual.* Institut für Machinelle Sprachverarbeitung, Germany.

Collins, M. 1997. Three generative, lexicalized models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, Bergen, Norway. 16–23.

Copestake, A. and Briscoe, E. J. 1991. Lexical operations in unification based framework. In *Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation*, Berkeley, California. 88–101.

Cover, T. M. and Thomas, J. A. 1991. *Elements of Information Theory.* John Wiley and Sons, Inc., New York.

Cunningham, H., Gaizauskas, R., and Wilks, Y. 1995. *A General Architecture for Text Engineering (GATE) - A New Approach to Language R&D.* Research memo CS-95-21, Department of Computer Science, University of Sheffield, UK.

Daelemans, W., Gazdar, G., and de Smedt, K. 1992. Inheritance in natural language processing. *Computational Linguistics* 18.2: 205–218.

Dang, H. T., Kipper, K., Palmer, M., and Rosensweig, J. 1998. Investigating regular sense extensions based on intersective Levin classes. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International*

Conference on Computational Linguistics, Montreal, Canada. 293–299.

Dorr, B. J. 1997. Large-scale dictionary construction for foreign language tutoring and interlingual machine translation. *Machine Translation* 12.4: 271–325.

Dorr, B. J. and Jones, D. 1996a. Acquisition of semantic lexicons: using word sense disambiguation to improve precision. In *Proceedings of the ACL Workshop on Breadth and Depth of Semantic Lexicons*, Santa Cruz, California. 42–50.

Dorr, B. J. and Jones, D. 1996b. Role of word sense disambiguation in lexical acquisition: predicting semantics from syntactic cues. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, Denmark. 322–333.

Dowty, D. 1991. Thematic proto-roles and argument selection. *Language* 67(3): 547–619.

Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19.1: 61–74.

EAGLES. 1996. *Recommendations on Subcategorization.* http://www.ilc.pi.cnr.it/pub/eagles/lexicons.

Elworthy, D. 1994. Does Baum-Welch re-estimation help taggers?. In *Proceedings of the 4th Conference on Applied* NLP, Stuttgart, Germany. 53–58.

Ersan, M. and Charniak, E. 1996. A statistical syntactic disambiguation program and what it learns. In Wermter, S., Riloff, E., and Scheler, G. eds. *Connectionist, Statistical and Symbolic Approaches in Learning for Natural Language Processing.* Springer-Verlag, Berlin: 146–157.

Evans, R. and Kilgarriff, A. 1995. MRDs, standards, and how to do lexical engineering. In *Proceedings of the 2nd Language Engineering Convention*, London, England. 125–134.

Fellbaum, C. 1999. The organization of verbs and verb concepts in a semantic net. In Saint-Dizier, P. eds. *Predicative Forms in Natural Language and in Lexical Knowledge Bases.* Kluwer Academic Publishers, Netherlands: 93–110.

Fillmore, C. J. 1968. The test for case. In Bach, E. and Harms, R. T. eds. *Universals in Linguistic Theory.* Holt, Rinehart and Winston, New York: 1–88.

Finch, S. and Chater, N. 1991. A hybrid approach to the automatic learning of linguistic categories. *Quarterly Newsletter of the Society for the Study of Artificial Intelligence and Simulation of Behaviour* 78: 16–24.

Francis, W. N. and Kučera, H. 1989. *Manual of Information to Accompany a Standard Corpus of Present-Day Edited American English, for use with Digital Computers (corrected and revised edition).* Department of Linguistics, Brown University.

Gahl, S. 1998. Automatic extraction of subcorpora based on subcategorization frames from a part-of-speech tagged corpus. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Canada. 428–432.

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

Garnsey, S. M., Lotocky, M. A., Pearlmutter, N. J., and Myers, E. M. 1997. *Argument Structure Frequency Biases for 1000 Sentence-Complement-Taking Verbs.* Unpublished manuscript. University of Illinois at Urbana-Champaign.

Garside, R., Leech, G., and Sampson, G. 1987. *The Computational Analysis of English: A Corpus-Based Approach.* Longman, London.

Gazdar, G. 1996. Paradigm merger in natural language processing. In Milner, R. and Wand, I. eds. *Computing Tomorrow: Future Research Directions in Computer Science.* Cambridge University Press, Cambridge: 88–109.

Gazdar, G., Klein, E., Pullum, G. K., and Sag, I. A. 1985. *Generalized Phrase Structure Grammar.* Blackwell, Oxford and Harvard University Press, Cambridge.

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

Goldberg, A. 1994. *A Construction Grammar Approach to Argument Structure.* Chicago University Press, Chicago.

Gorrell, G. 1999. *Acquiring Subcategorisation from Textual Corpora.* MPhil dissertation, University of Cambridge, UK.

Grimshaw, J. B. 1990. *Argument Structure.* MIT Press, Cambridge, Massachusetts.

Grishman, R., Macleod, C., and Meyers, A. 1994. Comlex syntax: building a computational lexicon. In *Proceedings of the International Conference on Computational Linguistics, COLING-94*, Kyoto, Japan. 268–272.

Grishman, R., Macleod, C., and Sterling, J. 1992. Evaluating parsing strategies using standardized parse files. In *Proceedings of the ACL Conference on Applied Natural*

Language Processing, Trento, Italy. 156–161.

Gruber, J. 1976. *Studies in Lexical Relations.* PhD thesis, MIT.

Guerssel, M. 1986. *On Berber verbs of change: a study of transitivity alternations.* Lexicon Project Working Papers 9, Center for Cognitive Science, MIT, Cambridge, Massachusetts.

Hajič, J. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In Hajicova, E. eds. *Issues of Valency and Meaning.* Karolinum, Praha: 106–132.

Hale, K. and Keyser, S. J. 1993. *On Argument Structure and Lexical Expression of Syntactic Relations.* MIT Press, Cambridge, Massachusetts.

Hindle, D. and Rooth, M. 1993. Structural ambiguity and lexical relations. *Computational Linguistics* 19.2: 103–120.

Hornby, A. S. 1989. *Oxford Advanced Learner's Dictionary of Current English.* Oxford University Press, Oxford.

Hudson, R. A. 1995. Identifying the linguistic foundations of lexical research and dictionary design. In Walker, D. E., Zampolli, A., and Calzolari, N. eds. *Automating the Lexicon: Research and Practice in a Multilingual Environment.* Oxford University Press, Oxford: 21–51.

Ide, N. and Veronis, J. 1995. *Text Encoding Initiative: Background and Context.* Kluwer, Dordrecht.

Inui, K., Sornlertlamvanich, V., Tanaka, H., and Tokunaga, T. 1997. A new formalization of probabilistic GLE parsing. In *Proceedings of the 5th ACL/SIGPARSE International Workshop on Parsing Technologies*, Bergen, Norway. 123–134.

Jackendoff, R. 1977. *X-bar syntax.* MIT Press, Cambridge, Massachusetts.

Jackendoff, R. 1990. *Semantic Structures.* MIT Press, Cambridge, Massachusetts.

Joshi, A., Levy. L, and M. Takahashi 1975. Tree-Adjunct grammars. *Journal of Computer Systems Science* 10: 136–163.

Jurafsky, D. and Martin, J. H. 2000. *Speech and Language Processing.* Prentice Hall.

Justeson, J. S. and Katz, S. M. 1995. Principled disambiguation: discriminating adjective senses with modified nouns. *Computational Linguistics* 17: 1–19.

Kalbfleisch, J. G. 1985. *Probability and Statistical Inference.* Volume 2, Second Edition. Springer-Verlag.

Katz, S. M. 1987. Estimation of probabilities from sparse data for the language model component of speech recogniser. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35.3: 400–401.

Katz, J. and Fodor, J. 1964. The structure of a semantic theory. In Fodor, J. and Katz, J. eds. *The Structure of Language.* Prentice Hall: 479–518.

Keller, F., Corley, M., Crocker, M. W., and Trewin, S. 1999. Gsearch: A tool for syntactic investigation of unparsed corpora. In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora*, Bergen, Norway. 56–63.

Kilgarriff, A. 2001. Comparing corpora. *International Journal of Corpus Linguistics. In Press.*

Kilgarriff, A. and Rosenzweig, J. 2000. *English Senseval: Report and Results.* ITRI, Brighton, UK.

Klavans, J. L. and Resnik, P. (eds.) 1996. *The Balancing Act.* The MIT press, Cambridge, Massachusetts.

Korhonen, A. 1998. Automatic extraction of subcategorization frames from corpora. In *Proceedings of the ESSLLI 98*, Saarbrücken, Germany. 49-56.

Korhonen, A. 2000. Using semantically motivated estimates to help subcategorization acquisition. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, Hong Kong. 216-223.

Korhonen, A., Gorrell, G., and McCarthy, D. 2000. Statistical filtering and subcategorization frame acquisition. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, Hong Kong. 199-205.

Lapata, M. 1999. Acquiring lexical generalizations from corpora: A case study for diathesis alternations. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, Maryland, USA. 397–404.

Lapata, M. 2000. *The Acquisition and Modeling of Lexical Knowledge.* PhD thesis, University of Edinburgh.

Lapata, M. and Keller, F. 1998. *Corpus Frequency as a Predictor of Verb Bias.* A poster presented at AMLAP-98, Freiburg.

Lapata, M., Keller, F., and Schulte im Walde, S. 2001. Verb frame frequency as a predictor of verb bias. *Journal of Psycholinguistic Research. In Press.*

Laplance, P. S. 1995. *Philosophical Essay On Probabilities.* Springer-Verlag.

Lascarides, A., Copestake, A., and Briscoe, E. J. 1996. Ambiguity and Coherence. *Journal of Semantics* 13: 41–65.

Lauer, M. 1995. Corpus statistics meet the noun compound. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, Massachusetts. 47–54.

Lee, L. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, Maryland, USA. 25–32.

Leech, G. 1992. 100 million words of English: the British National Corpus. *Language Research* 28(1): 1–13.

Levin, B. 1993. *English Verb Classes and Alternations.* Chicago University Press, Chicago.

Levin, B. and Rappaport Hovav, M. 1996. *Unaccusativity an the Syntactic-Lexical Semantics Interface.* MIT Press, Cambridge, Massachusetts.

Li, H. and Abe, N. 1995. Generalizing Case Frames Using a Thesaurus and the MDL Principle. In *Proceedings of the International Conference on Recent Advances in* NLP, Bulgaria. 239–248.

MacWhinney, B. 1996. The CHILDES System. *American Journal of Speech-Language Pathology* 5: 5–14.

Manning, C. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio. 235–242.

Manning, C. and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, Massachusetts.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19.2: 313–330.

Matthews, P. H. 1981. *Syntax.* Cambridge University Press, Cambridge, UK.

McArthur, T. 1981. *Longman Lexicon of Contemporary English.* Longman Group Ltd., UK.

McCarthy, D. and Korhonen, A. 1998. Detecting verbal participation in diathesis alternations. In *Proceedings of the 36th Annual Meeting of the Association for Com-*

putational Linguistics and 17th International Conference on Computational Linguistics, Montreal, Canada. 1493–1495.

McCarthy, D. 2001. *Lexical Acquisition at the Syntax-Semantics Interface: Diathesis Alternations, Subcategorization Frames and Selectional Preferences.* PhD thesis, University of Sussex.

McCarthy, M. (ed.) 1985. *Cambridge Word Selector.* Cambridge University Press, Cambridge, UK.

McNaught, J. 1990. Reusability of lexical and terminological resources; steps towards independence. In *Proceedings of the International Workshop on Electronic Dictionaries*, Kanagawa, Japan. 97–107.

Meyers, A., Macleod, C., and Grishman, R. 1994. *Standardization of the Complement Adjunct Distinction.* New York University, Ms.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. 1993. *Introduction to WordNet: An On-Line Lexical Database.* ftp//clarity.princeton.edu/pub/WordNet/5papers.ps.

Minnen. G., Carroll, J., and Pearce, D. 2001. Applied morphological processing of English. *Natural Language Engineering. In Press.*

Monachini, M. and Calzolari, N. 1996. *EAGLES Synopsis and Comparison of Morphosyntactic Phenomena Encoded in Lexicons and Corpora. A Common Proposal and Applications to European Languages.* Technical report 1996-TR-004. http://www.ercim.cnr.ilc.it.

Normier, B. and Nossin, M. 1990. GENELEX project: EUREKA for linguistic engineering. In *Proceedings of the International Workshop on Electronic Dictionaries*, Kanagawa, Japan. 63–70.

Pedersen, T. 1996. Fishing for exactness. In *Proceedings of the South-Central SAS Users Group Conference SCSUG-96*, Austin, Texas. 188–200.

Pereira, F., Tishby, N., and Lee, L. 1993. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association of Computational Linguistics*, Columbus, Ohio. 183–190.

Pereira, F. and Warren, D. 1980. Definite clause grammars for language analysis - a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence* 13.3: 231–278.

Pinker, S. 1989. *Learnability and Cognition: The Acquisition of Argument Structure.* MIT Press, Cambridge, Massachusetts.

Pirrelli, V., Ruimy, N., and Montemagni, S. 1994. *Lexical Regularities and Lexicon Compilation: Argument Structure Alternations of Italian Verbs.* Technical report 36, ACQUILEX-II.

Pollard, C. and Sag, I. A. 1987. *Information-based Syntax and Semantics Volume 1: Fundamentals.* CSLI, Stanford.

Pollard, C. and Sag, I. A. 1994. *Head-driven Phrase Structure Grammar.* Chicago University Press, Chicago.

Poznanski, V. and Sanfilippo, A. 1995. Detecting dependencies between semantic verb subclasses and subcategorization frames in text corpora. In Boguraev, B. and Pustojevsky, J. eds. *Corpus Processing for Lexical Acquisition.* MIT Press, Cambridge, Massachusetts: 175–190.

Price, P. 1996. Combining linguistic with statistical methodology in automatic speech understanding. In Klavans, J. L. and Resnik, P. eds. *The Balancing Act.* The MIT Press, Cambridge, Massachusetts: 112–148.

Procter, P. 1978. *Longman Dictionary of Contemporary English.* Longman, England.

Pustejovsky, J. 1991. The generative lexicon. *Computational Linguistics* 17.3: 409–441.

Resnik, P. 1993. *Selection and Information: A Class-Based Approach to Lexical Relationships.* PhD thesis, University of Pennsylvania.

Resnik, P. 1997. Selectional preference and sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why What and How?*, Maryland, USA. 52–57.

Ribas, F. 1995. *On Acquiring Appropriate Selectional Restrictions from Corpora Using a Semantic Taxonomy.* PhD thesis, University of Catalonia.

Rissanen, J. 1978. Modeling by shortest data description. *Automatica* 14: 465–471.

Roland, D. and Jurafsky, D. 1998. How verb subcatecorization frequencies are affected by corpus choice. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Canada. 1117-1121.

Roland, D. and Jurafsky, D. 2001. Verb sense and verb subcategorization probabilities. In Stevenson, S. and Merlo, P. eds. *The Lexical Basis of Sentence Processing: Formal, Computational, and Experimental Issues.* Jon Benjamins, Amsterdam: To appear.

Roland, D., Jurafsky, D., Menn, L., Gahl, S., Elder, E., and Riddoch, C. 2000. Verb subcatecorization frequency differences between business-news and balanced corpora. In *Proceedings of the ACL Workshop on Comparing Corpora*, Hong Kong. 28–34.

Rudanko, J. 1989. *Complementation and Case Grammar: A Syntactic and Semantic Study of Selected Patterns of Complementation in Present-Day English.* State University of New York Press, Albany.

Sampson, G. 1995. *English for the Computer.* Oxford University Press, Oxford, UK.

Sanfilippo, A. 1990. *Grammatical Relations, Thematic Roles and Verb Semantics.* PhD thesis, University of Edinburgh.

Sanfilippo, A. 1993. LKB encoding of lexical knowledge. In Briscoe, T., de Paiva, V., and Copestake, A. eds. *Inheritance, Defaults, and the Lexicon.* Cambridge University Press, Cambridge, UK: 190–222.

Sanfilippo, A. 1994. Word knowledge acquisition, lexicon construction and dictionary compilation. In *Proceedings of the International Conference on Computational Linguistics, COLING-94*, Kyoto, Japan. 273–277.

Sanfilippo, A. and Poznanski, V. 1992. The acquisition of lexical knowledge from combined machine-readable sources. In *Proceedings of the ACL Conference on Applied Natural Language Processing*, Trento, Italy. 80–87.

Sarkar, A. and Zeman, D. 2000. Automatic extraction of subcategorization frames for Czech. In *Proceedings of the 19th International Conference on Computational Linguistics*, Saarbrücken, Germany. 691–697.

Schulte im Walde, S. 2000. Clustering verbs automatically according to their alternation behaviour. In *Proceedings of the 18th International Conference on Computational Linguistics*, Saarbrücken, Germany. 747–753.

Schütze, H. 1992. Dimensions of meaning. In *Proceedings of the Supercomputing*, Los Alamitos, California. 787–796.

Schütze, H. 1993. Part-of-speech induction from scratch. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio. 251–258.

Shieber, S. 1984. The design of a computer language for linguistic information. In *Proceedings of the International Conference on Computational Linguistics*, Stanford, California. 362–366.

Shieber, S. 1986. *An Introduction to Unification-based Approaches to Grammar.* Chicago University Press, Chicago.

Siegel, S. and Castellan, N. J. (ed.). 1988. *Non-Parametric Statistics for the Behavioural Sciences.* McGraw-Hill, New York.

Sinclair, J. M. (ed.). 1987. *Collins Cobuild English Language Dictionary.* Collins, London.

Somers, H. L. 1984. On the validity of the complement-adjunct distinction in valency grammar. *Linguistics* 22: 507–520.

Spearman, C. 1904. The proof and measurement of association between two things. *American Journal of Psychology* 15: 72–101.

Stede, M. 1998. A generative perspective on verb alternations. *Computational Linguistics* 24.3: 401–430.

Stevenson, S. and Merlo, P. 1999. Automatic verb classification using distributions of grammatical features. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, Bergen, Norway. 45–52.

Taylor, L. and Knowles, G. 1988. *Manual of Information to Accompany the SEC Corpus: the Machine-Readable Corpus of Spoken English.* University of Lancaster, UK, Ms.

Ushioda, A., Evans, D., Gibson, T., and Waibel, A. 1993. The automatic acquisition of frequencies of verb subcategorization frames from tagged corpora. In Boguraev, B. and Pustejovsky, J. eds. *SIGLEX ACL Workshop on the Acquisition of Lexical Knowledge from Text.* Columbus, Ohio: 95–106.

Verspoor, C. M. 1997. *Contextually-Dependent Lexical Semantics.* PhD thesis, University of Edinburgh.

Wechsler, S. 1995. *The Semantic Basis of Argument Structure.* CSLI Publications, Stanford, California.

Wilks, Y. 1986. An intelligent analyzer and understander of English. In Grosz, B., Sparck-Jones K., and Webber, B. eds. *Readings in Natural Language Processing.* Morgan Kaufmann, California: 264–274.

Zeevat, H., Klein, E., and Calder, J. 1987. An introduction to Unification Categorial Grammar. In Haddock, J. N., Klein E., and Morrill, G. eds. *Edinburgh working papers in cognitive science, Vol. 1: Categorial Grammar, Unification Grammar and Parsing*. University of Edinburgh, UK: 195–222.

Zernik, U. 1989. Lexical acquisition: learning from corpora by capitalising on lexical categories. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Detroit, USA. 1556–1564.