Automatic Extraction of Large-Scale Multilingual Lexical Resources

Ruth O'Donovan

B.A., M.Sc.

A dissertation submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy

to the

DCU

Dublin City University

School of Computing

Supervisors: Prof. Josef van Genabith
Dr. Andy Way

February 2006

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work

Signed Ruth O'Donovan

(Ruth O'Donovan)

Student ID 52178218

Date February 2006

Contents

Abstract

In this thesis, I present a methodology for treebank- or parser-based acquisition of lexical resources, in particular subcategorisation frames. The method uses an automatic Lexical Functional Grammar (LFG) f-structure annotation algorithm (Cahill et al., 2002a, 2004a; Burke et al., 2004b) and has been applied to the Penn-II and Penn-III treebanks (Marcus et al., 1994) with a total of about 1.3 million words as well as to (a subset of) the British National Corpus (Bernard, 2002) with about 90 million words.

I extract abstract syntactic function-based subcategorisation frames (LFG semantic forms), traditional CFG category-based subcategorisation frames as well as mixed function/category-based frames, with or without preposition information for obliques and particle information for subcategorised particles. The approach distinguishes between active and passive frames, and reflects the effects of long-distance dependencies (LDDs) in the source data structures. Frames are associated with conditional probabilities, facilitating the optimisation of the extracted lexicon for quality or coverage through filtering. In contrast to many other approaches, subcategorisation frame types are not predefined but acquired from the source data.

I carried out large-scale evaluations of the complete set of forms extracted against the COMLEX and OALD resources. To my knowledge, this is the largest and most complete evaluation of subcategorisation frames for English. The parser-based system is also evaluated against Korhonen (2002) with a statistically significant improvement over the previous best score.

The automatic annotation methodology, as well as the grammar and lexicon extraction techniques for English have been successfully migrated to Spanish, German and Chinese treebanks despite typological differences and variations in treebank encoding. I believe that this approach provides an attractive and efficient multilingual grammar and lexicon development paradigm.

Acknowledgements

The creation of this thesis was facilitated in various ways by a number of people. I want to acknowledge their efforts here. First, I wish to express my gratitude to my supervisors Josef van Genabith and Andy Way. Josef has been an inspiration. His enthusiasm often revived mine, and his encouragement and feedback were crucial to completing this work. Andy has constantly provided me with practical advice and insightful, intelligent comments on this thesis. More generally, I would like to thank both Josef and Andy for accepting me in DCU when I chose to transfer here. From the outset, they have encouraged me to achieve more than I thought possible during the course of my study and for that I am truly thankful.

My time at DCU has also been enriched, both academically and socially, by the members of the NCLT. I have really enjoyed being surrounded by Declan, Mary, Michelle, Nano and the other postgrads. Mick and Aoife in particular have been ideal colleagues. Aoife has always been patient and helpful, and I have learned so much from her. Mick has been a fantastic support both academically and personally. Writing our theses concurrently made the process so much easier! I thank him for his patience, his football knowledge, the coffee breaks and always being the voice of reason.

There are a number of people outside of DCU who have made the last three (busy) years of my life infinitely more enjoyable. I would like to thank all the expats in Dublin: Lena (my wonderful flatmate who always made sure I was eating well), Maribel, Siobhan, Alan, Liam, Nessa, Pedz, Kitty, Bryan, Tracy, Fergal, Dee and Shane (especially for the Wednesday lunches!). I am also grateful to my friends outside the pale: Aileen, Fiona, Soc, John, Patch, Eileen, Jill and Bren who I miss so much, and Allie who has a been a great friend for such a long time.

I would like to thank my brothers, Gary, David and Brian, along with their ladies, Shelly and Jo. I am especially grateful to my wonderful parents, Mary and Jim, for their love, encouragement and constant support of their "eternal student" daughter. Finally, very special thanks go to Mark for, most notably, answering my statistics questions, listening to me, understanding me, playing scrabble with me, making me laugh and being my best friend.

List of Tables

List of Figures

List of Acronyms

AI	Artifical Intelligence
AVM	Attribute-Value Matrix
BNC	British National Corpus
BHT	Binomial Hypothesis Test
CCG	Combinatory Categorial Grammar
CF-PSG	Context-Free Phrase Structure Grammar
CFG	Context-Free Grammar
FU	Functional Uncertainty
GF	Grammatical Function
HPSG	Head-driven Phrase Structure Grammar
LDD	Long-Distance Dependency
LDOCE	Longman's Dictionary of Concise English
LFG	Lexical-Functional Grammar
LHS	Left Hand Side
MRD	Machine Readable Dictionary
NLP	Natural Language Processing
OALD	Oxford Advanced Learners' Dictionary
PCFG	Probabilistic Context-Free Grammar
POS	Part Of Speech
RHS	Right Hand Side
SVO	Subject-Verb-Object
TAG	Tree Adjoining Grammar
VSO	Verb-Subject-Object
WSJ	Wall Street Journal

Chapter 1

Introduction

This thesis describes a large-scale, robust, flexible methodology for the automatic acquisition of lexical resources from both treebanks and raw text suitable for application to typologically diverse languages. In the introduction I motivate the work and outline the nature of the information induced as well as its practical application.

While the methodology presented here can capture lexical information for headwords of any syntactic category, I focus on the extraction of verbal lexical resources. In particular, the system induces information about the subcategorisation tendencies of verbs, the syntactic or functional arguments required by a verb to form a grammatical construction. A grammatical construction may additionally contain adjuncts whose omission, while (potentially) affecting the meaning of an utterance, does not result in an ungrammatical construction.

The manual construction of a dictionary or lexicon is a time-consuming and expensive process. In fact, the manual construction of lexical resources constitutes an instance of the "knowledge acquisition bottleneck" familiar from other knowledge-intensive approaches to natural language processing (NLP) and artificial intelligence (AI). Lexicons are important linguistic resources in NLP applications.[1] However, existing hand-crafted machine-readable dictionaries (MRDs) such as COMLEX (Grishman et al., 1994; Macleod et al., 1994) and the OALD (Hornby, 1980) are limited in their usefulness for a number of reasons. The static nature of manually produced lexicons fails to capture the productivity

[1] And, of course, valuable resources for human users.

of natural language, resulting in limited coverage when faced with new material both in terms of verbs for which entries exist and how these verbs interact with other syntactic elements to form grammatically acceptable structures. Second, MRDs do not always make a distinction between arguments and adjuncts. COMLEX is a subcategorisation lexicon where the argument/adjunct distinctions are made following a set of predefined guidelines (Meyers et al., 1994). The OALD represents verbal behaviour in terms of verb patterns (which may potentially contain adjuncts) rather than subcategorisation frames. Third, hand-crafted lexicons do not contain any information about the relative frequency with which verbs occur with certain frames or patterns. With recent approaches to NLP becoming increasingly empirical in nature, this information is very valuable. Finally there is a dearth of MRDs for languages other than English. For all of these reasons, extensive research has been carried out on the automatic acquisition of lexical resources from corpora. A large body of work (Brent, 1993; Manning, 1993; Ushioda et al., 1993; Briscoe and Carroll, 1997; Carroll and Rooth, 1998; Schulte im Walde, 2002a) takes raw text as input, preprocesses it to varying degrees, hypothesises (pre-defines) a set of verb-frame associations and refines the hypotheses using statistical filtering. A second body of work (Hockenmaier et al., 2004; Miyao et al., 2004) takes hand-corrected, parse-annotated data (a treebank) as input, enriches the input data using linguistic generalisations and extracts a set of verb-frame associations which may be filtered.

This thesis presents a method for extracting rich grammatical-function-based lexical resources from both treebank and raw data. Treebank-based subcategorisation frame extraction holds the promise of high-precision results as the input data (parse trees) are usually hand-corrected. The major drawback is the limited size of input data available. By contrast, parser-mediated acquisition from raw text opens up the possibility of extraction from very large data-sets with a potential quality decrease due to noise introduced by the parser. The first step in the acquisition of the lexical resources is the automatic annotation of basic parse trees (either manually or automatically produced) with Lexical Functional Grammar (LFG: (Kaplan and Bresnan, 1982; Bresnan, 2001; Dalrymple, 2001)) f-structure equations. The automatic f-structure annotation algorithm (Cahill et al., 2002a, 2004a; Burke et al., 2004b) exploits categorial, configurational, head and trace (in the case of treebank trees) information to annotate trees with syntactic functional information approximating to basic predicate-argument or dependency structures. In theoretical LFG the predicate value in an f-structure is a *semantic form* (a lemma followed by an argument list e.g. FOCUS$\langle(\uparrow \text{ SUBJ})(\uparrow \text{ OBL}_{on})\rangle$) the value of which constrains the well-formedness of the f-structure. In the automatically produced f-structures of Cahill et al. (2002a, 2004a); Burke et al. (2004b), the predicate value is a lemma. Having carried out extensive evaluations to ensure f-structure quality, the extraction methodology presented in this thesis reverse-engineers semantic forms corresponding to the predicate lemma values in the f-structures. While the arguments of the semantic forms are expressed in terms of LFG functions, categorial information is also extracted allowing the expression of arguments in terms of syntactic category or a combination of syntactic category/grammatical function. Verbs occurring with subcategorised-for PPs and particles are parameterised for specific prepositions and particles. The extracted frames reflect long distance dependencies (LDDs) in the source data and the system differentiates between frames used actively and passively. The relative frequency of the frames is calculated and used in statistical filtering.

An important aspect of the approach presented in this thesis is its robustness and scalability. We first apply it to the Penn-III Treebank comprising 50,000 Wall Street Journal (WSJ) sentences and a parse-annotated subset (25,000 sentences) of the Brown Corpus, extracting over 20,000 semantic form types. In contrast to similar approaches by Hockenmaier et al. (2004) and Miyao et al. (2004), we do not include a treebank preprocessing step (including clean-up and binarisation) but use the treebank as is.

As large-scale extraction is important, it is necessary to carry out a large-scale evaluation to ensure the quality of the extracted resources. Previous approaches to lexical resource evaluation have generally focused on the evaluation of a small number of frequently occurring verbs against either a manually constructed or externally available gold standard. While there are advantages and disadvantages to both types of gold standard, I chose to evaluate against COMLEX and the OALD. To facilitate this I developed an automatic mapping between the gold-standard frames and the induced frames. For Penn-III, I evaluate approximately 3,500 actively used verb lemmas against both the OALD and

COMLEX, in both cases outperforming the baseline.[2] To our knowledge, there has been no larger evaluation of an English lexical extraction system in terms of verb number. In addition, I evaluate the passive frames induced by our system by automatically applying Lexical Redundancy Rules (Kaplan and Bresnan, 1982) to the gold-standard COMLEX frames to automatically produce a passive gold-standard.

Korhonen (2002) points out that the real value of automatically induced lexical resources may be demonstrated by the improvement they effect in a practical NLP task. This thesis is part of a larger project focused on the automatic acquisition of LFG-based grammar and lexical resources. The lexical resources (both active and passive) described here along with their conditional probabilities are incorporated into the parsing architecture described in (Cahill et al., 2004b) for the resolution of long-distance dependencies (LDDs). I present results which illustrate the improvement in parser performance effected by using the automatically acquired lexical information in this way.

I then present a modular extension to the basic lexical extraction architecture in order to acquire lexical information from raw text. The text is first tagged and parsed using a choice of two taggers, MXPOST (Ratnaparkhi, 1996) and TreeTagger (Schmid, 1994), and a choice of three history-based statistical parsers Charniak (2000), Collins (1999) Models 2 and 3, and Bikel (2002). All parsers are trained on Sections 02-21 of the WSJ. Normally Penn-II functional tags are stripped from the trees prior to training. In the case of Bikel (2002) which is retrainable, we also train the parser to retain Penn-II tags. Next the automatically produced parse trees for unseen text are automatically annotated using the f-structure annotation algorithm. Using the annotations, the constraint solver produces a set of 'proto' f-structures (with unresolved LDDs). We run a first lexical induction pass, extracting 'safe' semantic forms only from those sub-f-structures not containing an unresolved LDD. The extracted semantic forms and their frequencies are combined with those acquired from the original treebank trees in Sections 02-21 of the WSJ (the training set for the parsers) and conditional probabilities are re-calculated. This resource is then used to resolve the LDDs to produce 'proper' f-structures for unseen text from which a final complete set of semantic forms is extracted. This extension to the basic architecture allows

[2] A baseline score is computed by automatically assigning two most frequent frames, transitive and intransitive, to each verb and evaluating this 'artificial' lexicon against the gold standard.

us to scale the lexicon extraction to 90 million words of raw BNC text. We evaluate the induced lexicon against the OALD and COMLEX using the automated mapping procedure as before. Again, the baseline is exceeded in each case.

Importantly, extending our extraction architecture to deal with raw text affords us the opportunity to directly compare our methodology with that of a state-of-the-art lexical extraction system (Korhonen, 2002). Using the evaluation resources of Korhonen (2002) which include a 65,000 word subcorpus of the BNC, evaluation software, a gold standard of 30 test verbs as well as the files to replicate her best results, we compare the performance of the two systems. The LFG-based extraction system achieves an f-score of 76.16% which is statistically significantly better than the replicated score of Korhonen (2002).

I carry out experiments to directly compare parser- and treebank-based lexical acquisition and study the effect of sentence length on the coverage and quality of the induced resources.

Substantial treebanks or dependency banks are becoming available for languages other than English such as Chinese, German, Spanish, Catalan, French and Japanese. We show how the automatic annotation as well as the grammar and lexical induction approaches presented for English may be migrated to other typologically different languages such as German (Cahill et al., 2003, 2005b), Chinese (Burke et al., 2004c) and Spanish (O'Donovan et al., 2005b). In this thesis I focus on the work carried out for Spanish and the Cast3LB Treebank (Civit, 2003). I extract PCFG-based LFG approximations and parse unseen Spanish text into f-structures, achieving an f-score of 73.20% (preds-only) against a manually constructed gold standard of 100 f-structures. In addition, I extract 3136 semantic form types for 1401 verb lemmas.

The work reported in this thesis is part of a larger project on the treebank-based acquisition of wide-coverage, deep, robust, probabilistic Lexical-Functional Grammar resources with Michael Burke in charge of the development of the English f-structure annotation algorithm and Aoife Cahill in charge of the parser and generator resources. In the dissertation, in close cooperation with Michael Burke and Aoife Cahill, I use and modify both the annotation algorithm and the parsing resources. I have adopted the following convention: I will use 'I' for my independent contribution and I will use 'we' for cases where I have been the lead researcher for subcategorisation frame extraction in a team that includes Michael Burke and Aoife Cahill.

This thesis is structured as follows:

Chapter 2 presents various approaches to the automatic extraction of lexical resources organised based on input data.

Chapter 3 describes the basic lexical extraction methodology. It includes an overview of the LFG formalism and the automatic f-structure annotation algorithm.

Chapter 4 describes treebank-based lexical extraction using the methodology described in Chapter 3 and taking the Penn-III Treebank as input. It includes a detailed evaluation of the extracted resources against the OALD and COMLEX as well as a task-based evaluation. I also present a quantitative evaluation illustrating the coverage of the extracted resources on new text as well as the lexical accession rate of the system.

Chapter 5 presents the architecture for parser-based lexical extraction from the BNC based on the original methodology presented in Chapter 3. It includes a set of small-scale experiments on the WSJ motivating this work based on the quality of the output. The chapter presents an evaluation comparing our work to that of Korhonen (2002) using her freely available evaluation resources. I carry out a large-scale evaluation against the OALD and COMLEX using the automatic mapping described in Chapter 4. I measure the effect of sentence length on lexicon quality and coverage. Finally I illustrate the lexical accession rates of the system on the BNC.

Chapter 6 describes the migration of the methodology developed for English to Spanish and the Cast3LB treebank. I evaluate the automatic annotation as well as the parsing of new text against a manually constructed gold standard. I also present the details of a set of automatically extracted set of Spanish semantic forms and briefly discuss similar work carried out for German and Chinese.

Chapter 7 concludes and outlines areas of future work.

Chapter 2

Background and Related Work

2.1 Introduction

The encoding of verb subcategorisation properties is an essential step in the construction of computational lexicons for tasks such as parsing, generation and machine translation. Creating such a resource by hand is time-consuming, error-prone, requires considerable linguistic expertise and is rarely if ever complete. In addition, a hand-crafted lexicon cannot be easily adapted to specific domains or account for linguistic change. Accordingly, many researchers have attempted to construct lexical resources automatically. In this chapter, I discuss approaches to CFG-based subcategorisation frame extraction as well as attempts to induce lexical resources which comply with specific linguistic theories, or express information in terms of more abstract predicate-argument relations. The approaches are organised based on the nature of the data from which the lexical information is extracted. Either raw text (which may optionally be automatically tagged, chunked or parsed) or manually parse-annotated treebank data are used as input to the subcategorisation frame extraction system.

2.2 Verbal Subcategorisation

A subcategorisation frame encodes the arguments required by a verb to form a grammatical construction. Many verbs, particularly the more commonly occurring verbs, are associated with more than one such frame. For example, the verb `write` may be used intransitively (subcategorising for a single subject argument), transitively (subcategorising for a subject and an object) or ditransitively (subcategorising for a subject, a direct object and an indirect object). Statistical parsers use this information for disambiguation but they require comprehensive lists of lexical entries along with their associated frequencies to determine the most likely analysis. In practice, information about the frequencies of verb frame combinations can only deduced by automatically analysing large quantities of naturally occurring text.

An accurate parser is required for the automatic induction of subcategorisation information but paradoxically requires lexical information for optimal performance. Noisy frames introduced by the parser must be filtered statistically at the risk of rejecting rare but valid frames. Argument/adjunct distinctions are not always clear, particularly in the challenging area of prepositional phrase attachment. For all of these reasons, there is a ceiling on the performance of current approaches to subcategorisation frame acquisition which I hope address in this thesis.

2.3 Approaches to Lexical Acquisition

In keeping with the strands of work presented in this dissertation, I divide related approaches to lexical acquisition into two groups: approaches taking raw text as input and approaches taking pre-parsed and hand-corrected treebank data as their input. While the majority of the work presented here has been carried out for English, where possible we include details of the induction of lexical information for other languages. Evaluation of the methodologies is not discussed here (cf. Section 4.3).

2.3.1 Raw Text-Based Approaches

Typically in the approaches based on raw text, a number of subcategorisation patterns are predefined, a set of verb-subcategorisation frame associations are hypothesised from the data and statistical methods are applied to reliably select hypotheses for the final lexicon. Although I group all of these approaches together based on their source data, they vary based on the level of automatic preprocessing of this data prior to lexical extraction.

Brent (1993) relied on unambiguous morphosyntactic cues in the untagged Brown Corpus as indicators of 6 predefined subcategorisation frames, not parameterised for specific prepositions. Brent used hypothesis testing on binomial frequency data to statistically filter the induced frames. This approach is very conservative resulting in a very accurate lexicon with very limited coverage.

Ushioda et al. (1993) run a finite-state NP parser on a POS-tagged version of the Wall Street Journal corpus to calculate the relative frequency of the same 6 subcategorisation verb classes. The experiment is limited by the fact that all prepositional phrases are treated as adjuncts. Ushioda et al. (1993) employ an additional statistical method based on log-linear models and Bayes' Theorem to filter the extra noise introduced by the parser and were the first to induce relative frequencies for the extracted frames.

Manning (1993) attempts to improve on the approach of Brent (1993) by passing raw text through a stochastic tagger and a finite-state parser (which includes a set of simple rules for subcategorisation frame recognition) in order to extract verbs and the constituents with which they co-occur. He assumes 19 different subcategorisation frame definitions and the extracted frames include details of specific prepositions. The extracted frames are noisy due to parser errors and so are filtered using the Binomial Hypothesis Test (BHT), following Brent (1993). Applying his technique to approximately 4 million words of New York Times newswire, Manning acquired 4900 verb-subcategorisation frame pairs for 3104 verbs, an average of 1.6 frames per verb. The lexical entries are not associated with relative frequencies.

In contrast to the chunker-based approaches described above, Carroll and Rooth (1998) use a hand-written head-lexicalised context-free grammar to compute the probability of particular subcategorisation patterns in a given text corpus. The approach is iterative with the aim of estimating the distribution of subcategorisation frames associated with a particular predicate. They perform a mapping between their frames and those of the OALD resulting in 15 frame types. These do not contain details of specific prepositions.

Briscoe and Carroll (1997) predefine 163 verbal subcategorisation frames, obtained by manually merging the classes exemplified in the COMLEX (Macleod et al., 1994) and ANLT (Boguraev et al., 1987) dictionaries, and adding around 30 frames found by manual inspection. The frames incorporate control information and details of specific prepositions. The raw corpus data taken as input by the system is tagged, lemmatised and parsed using a robust statistical parser. Patterns are extracted from local parse trees and where possible mapped to the predefined subcategorisation frames. Briscoe and Carroll (1997) refine the BHT with *a priori* information about the probabilities of subcategorisation frame membership and use it to filter the induced frames. Recent work by Korhonen (2002) on the filtering phase of this approach uses linguistic verb classes (based on (Levin, 1993)) for obtaining more accurate back-off estimates for hypothesis selection resulting in improved performance.

Schulte im Walde (2002a,b) uses a head-lexicalised probabilistic context-free grammar similar to Carroll and Rooth (1998) to extract subcategorisation frames from a large German newspaper corpus from the 1990's. She predefines 38 distinct frame types, which contain maximally three arguments each and are made up of a combination of the following: nominative, dative and accusative noun phrases, reflexive pronouns, prepositional phrases, expletive *es*, subordinated non-finite clauses, subordinated finite clauses and copula constructions. The frames may optionally contain details of particular prepositional use. Unsupervised training is performed on a large German newspaper corpus and the resulting probabilistic grammar establishes the relevance of different frame types to a specific lexical head. Due to computing time constraints, she limits sentence length for grammar training and parsing. Sentences with length between 5 and 10 words were used to bootstrap the lexicalised grammar model. For lexicalised training, sentences of length between 5 and 13 words were used. The result is a subcategorisation lexicon for over 14,000 German verbs.

2.3.2 Treebank-Based Approaches

Approaches using treebank data as a source for subcategorisation information do not predefine the frames to be extracted but rather learn them from the data. Kinyon and Prolo (2002) describe a simple tool which uses fine-grained rules to identify the arguments of verb occurrences in the Penn-II Treebank. This is made possible by manual examination of over 150 different sequences of syntactic and functional tags in the treebank. For each sequence, it was decided whether it should be categorised as a modifier or argument. Arguments were then mapped to traditional syntactic functions. For example, the tag sequence NP-SBJ denotes a mandatory argument and its syntactic function is subject. In general, argumenthood was preferred over adjuncthoood. As (Kinyon and Prolo, 2002) does not include an evaluation, currently it is impossible to say how effective their technique is.

A substantial amount of work has been carried out on the extraction of TAG-, CCG- and HPSG-based formalism-specific lexical resources from the Penn-II Treebank. As these formalisms are fully lexicalised with an invariant (LTAG and CCG) or limited (HPSG) rule component, the extraction of a lexicon essentially amounts to the creation of a grammar. Chen and Vijay-Shanker (2000) explore a number of related approaches to the extraction of a lexicalised TAG from the Penn-II Treebank with the aim of constructing a statistical model for parsing. The extraction procedure utilises a head percolation table as introduced by Magerman (1994) in combination with a variation of the approach of Collins (1999) to the differentiation between complement and adjunct. This results in the construction of a set of lexically anchored elementary trees which make up the TAG in question. The number of frame types extracted (i.e. an elementary tree without a specific lexical anchor) ranged from 2366 to 8996. Xia (1999) presents a similar method for the extraction of a TAG from the Penn-II Treebank. The extraction procedure consists of three steps: firstly, the bracketing of the trees in the Penn-II treebank is corrected and extended based on the approaches of Magerman (1994) and Collins (1999). Then the elementary trees are read off in a quite straightforward manner. Finally any invalid elementary trees produced as a result of annotation errors in the treebank are filtered out using linguistic heuristics. The number of frame types extracted by Xia (1999) ranged from 3014 to 6099.

Hockenmaier et al. (2004) outline a method for the automatic annotation of the Penn-II treebank for the extraction of a large syntactic CCG lexicon. The approach of Hockenmaier et al. (2004) includes a preprocessing step during which POS-tag errors in the treebank are corrected and certain Penn-II analyses are altered to facilitate annotation. This preprocessing step is described in (Hockenmaier and Steedman, 2002). The algorithm annotates the nodes of the preprocessed trees with CCG categories in a top-down recursive manner. The first step is to label each node as either a head, complement or adjunct based on the approaches of Magerman (1994) and Collins (1999). The trees are then binarised. Finally, each node is assigned the relevant CCG category based on its constituent type and surface configuration. The algorithm handles 'like' coordination and exploits the traces used in the treebank in order to interpret long-distance dependencies (LDDs).

Miyao et al. (2004) and Nakanishi et al. (2004) describe a methodology for acquiring an English HPSG grammar from the Penn-II Treebank. Each tree in the treebank is annotated with partially-specified HPSG derivation trees in the following way: head/argument/modifier distinctions are made for each node in the tree based on (Magerman, 1994) and (Collins, 1999), the tree is then binarised, heuristics are applied to deal with phenomena such as LDDs and coordination, and to correct some errors in the treebank, and finally a HPSG category is assigned to each node in the tree in accordance with its CFG category. HPSG lexical entries are then automatically extracted from the annotated trees through the application of "inverse schemata".

With the increasing availability of treebanks for other languages, approaches similar to those described above have been applied to languages other than English. Sarkar and Zeman (2000) present an approach to learn previously unknown frames for Czech from the Prague Dependency Bank (Hajic, 1998). Czech is a freer word order language than English and so configurational information cannot be relied upon. In a dependency tree, the set of all dependents of the verb make up a so called 'observed frame', while a subcategorisation frame contains a subset of the dependents in the observed frame. Finding subcategorisation frames involves filtering adjuncts from the observed frame. This is achieved using three different hypothesis tests: BHT, log-likelihood ratio and t-score. The system learns 137 subcategorisation frames from 19,126 sentences for 914 verbs (those which occurred 5 times or more).

Marinov (2004) presents preliminary work on the automatic extraction of subcategorisation frames for Bulgarian from the BulTreeBank (Simov et al., 2002). In a similar way to that of Sarkar and Zeman (2000), his system collects both arguments and adjuncts. The Binomial Log Likelihood Ratio is used to filter incorrect frames. The BulTreebank trees are annotated with HPSG typed feature structure information and thus contain more detail than the dependency trees. The work done for Bulgarian is small-scale, however, as Marinov is working with a preliminary version of the treebank consisting of 580 sentences.

2.4 Summary

I have presented related approaches to lexical acquisition, organised by the type of input data to the extraction system: lexical acquisition from raw text (which may be automatically tagged, chunked or parsed) and acquisition from (manually parse-annotated) treebanks.

A number of approaches use raw corpus data, which is automatically tagged, chunked or fully parsed prior to subcategorisation frame extraction. Due to the margin of error associated with the lexical extraction compounded by each of these processing steps, a set of frames is predefined and only syntactic patterns mapping to those predefined frames are considered during extraction. The automatically induced lexical information is noisy and statistical filtering is used to increase accuracy.

In contrast, other approaches exploit 'gold standard' or manually parse-annotated data (treebanks). Due to the rich and accurate information provided by the annotation of the input data, frames may be learned rather than manually predefined. Often the information contained in the treebank annotation permits the extraction of richer frames containing functional and categorial information. Frames extracted from treebanks can reflect the effect of LDDs in the input strings. Despite the high quality of the input data, statistical filtering may be applied to increase lexical accuracy.

Raw text-based lexical acquisition (with optional automatic tagging, chunking or parsing preprocessing) has the advantage of supporting acquisition from very large data sets. The main disadvantage is the error margin or amount of noise introduced by each automatic (pre-)processing step. By contrast, treebank-based acquisition can avail of high quality and rich annotations but is limited to the size of the treebank.

In Chapter 3, I present a methodology which may take either treebank data or raw corpus text as input. Regardless of input type, the methodology produces rich functional and/or categorial-based subcategorisation information, reflecting LDDs and distinguishing between active and passive input. During the course of this dissertation I demonstrate the robustness and accuracy of this approach, as well as its suitability to multilingual lexical extraction.

Chapter 3

Methodology

3.1 Introduction

In modern syntactic theories (e.g. Lexical-Functional Grammar (LFG: (Kaplan and Bresnan, 1982; Bresnan, 2001; Dalrymple, 2001)), Head-Driven Phrase Structure Grammar (HPSG: (Pollard and Sag, 1994)), Tree-Adjoining Grammar (TAG: (Joshi, 1988)) and Combinatory Categorial Grammar (CCG: (Ades and Steedman, 1982))), the lexicon is the central repository for much morphological, syntactic and semantic information. Extensive lexical resources, therefore, are crucial in the construction of wide-coverage computational systems based on such theories. In this chapter, I present a methodology for the automatic extraction of lexical resources for use with an LFG-based parsing system. As input this method requires f-structures or an f-structure-annotated set of trees. The trees may be obtained from a manually constructed treebank or automatically produced by a parser. In either case, they are automatically annotated with f-structure information (Cahill et al., 2002a; Burke et al., 2004b). Lexical resources are then reverse-engineered from the resulting set of f-structures. The architecture is modular and flexible and ported successfully to other, typologically diverse languages (Cahill et al., 2003, 2005b; Burke et al., 2004c; O'Donovan et al., 2005b).

Section 3.2 provides a short description of the Lexical Functional Grammar formalism and justifies its use for this work. Section 3.3 presents our automatic f-structure annotation algorithm including a description of the architecture. Section 3.4 describes the lexical extraction algorithm, including specific extensions to the f-structure annotation algorithm. Much of the work presented in this chapter is described in (O'Donovan et al., 2004) and (O'Donovan et al., 2005b).

3.2 Lexical Functional Grammar

Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982; Bresnan, 2001; Dalrymple, 2001) is a member of the family of unification or constraint-based grammars. It posits minimally two levels of syntactic representation:[1] c(onstituent)-structure encodes details of language-specific surface syntactic constituency in the form of context-free trees. F(unctional)-structure expresses abstract syntactic information about predicate-argument-modifier relations and certain morphosyntactic properties such as tense, aspect and case, and is implemented in the form of recursive attribute-value matrices.

C-structure and f-structure representations are related in terms of "functional annotations" of the form $\uparrow$ ...=$\downarrow$... on tree nodes, i.e. attribute-value structure equations describing f-structures. This is exemplified by the analysis of the string '`The inquiry soon focused on the judge`' (wsj_0267_72) using the grammar in Figure 3.1 which results in the annotated c-structure and f-structure in Figure 3.2. Each node in the c-structure is annotated with f-structure equations e.g. ($\uparrow$ SUBJ)=$\downarrow$. The uparrows ($\uparrow$) point to the f-structure associated with the mother node and downarrows ($\downarrow$) to that of the local node. The ($\uparrow$ SUBJ)=$\downarrow$ on the NP-SBJ node in Figure 3.2 states that the f-structure associated with that node is the SUBJ of its mother node (in this case the S node). In a complete parse tree these $\uparrow$ $\downarrow$ meta variables are instantiated to unique tree node identifiers and a set of constraints is produced which if satisfiable generates an f-structure.

3.2.1 The LFG Lexicon

The value of the PRED attribute in an f-structure is a *semantic form*: $\Pi\langle gf_1,gf_2,\ldots,gf_n\rangle$ where Π is a lemma and *gf* a grammatical function. The semantic form provides a *frame type* $\langle gf_1,gf_2,\ldots,gf_n\rangle$ specifying the *governable grammatical functions* (or arguments) required by the predicate to form a grammatical construction. In Figure 3.1 the verb

[1] LFGs may also involve morphological and semantic levels of representation.

S →	NP-SBJ (↑ SUBJ)=↓	ADVP-TMP ↓ ∈ ↑ ADJ	VP ↑=↓
NP-SBJ →	DT (↑ SPEC DET)=↓	NN ↑=↓	
VP →	VBD ↑=↓	PP-CLR (↑ OBL)=↓	
ADVP-TMP →	RB ↑=↓		
PP-CLR →	IN ↑=↓	NP (↑ OBJ)=↓	
NP →	DT (↑ SPEC DET)=↓	NN ↑=↓	
focused	VBP	(↑ PRED)='FOCUS⟨(↑ SUBJ)(↑ OBL$_{on}$)⟩' (↑ TENSE)=PAST	
inquiry	NN	(↑ PRED)='INQUIRY' (↑ NUM)=SG (↑ PERS)=3	
judge	NNS	(↑ PRED)='JUDGE' (↑ NUM)=SG (↑ PERS)=3	
on	IN	(↑ PRED)='ON⟨(↑ OBJ)⟩'	
soon	RB	(↑ PRED)='SOON'	
the	DT	(↑ PRED)='THE'	

Figure 3.1: Sample LFG Rules and Lexical Entries

focus requires a subject and an oblique object introduced by the preposition on: FOCUS⟨(↑ SUBJ)(↑ OBL$_{on}$)⟩. The argument list can be empty as in the PRED value for judge in Figure 3.1. According to Dalrymple (2001), LFG assumes the following universally available inventory of grammatical functions: SUBJ(ect), OBJ(ect), OBJ$_\theta$, COMP, XCOMP, OBL(ique)$_\theta$, ADJ(unct), XADJ and POSS. OBJ$_\theta$ and OBL$_\theta$ cover families of grammatical functions indexed by their semantic role, represented by the θ subscript. This list of grammatical functions is divided into governable (subcategorisable) grammatical functions (*arguments*) and non-governable (non-subcategorisable) grammatical functions (*modifiers/adjuncts*), as summarised in Table 3.1.

A number of languages allow the possibility of object functions in addition to the primary OBJ, such as the second or indirect object in English. Oblique arguments are realised as prepositional phrases in English. COMP, XCOMP and XADJ are clausal functions which differ in the way in which they are controlled. A COMP is a *closed* function which

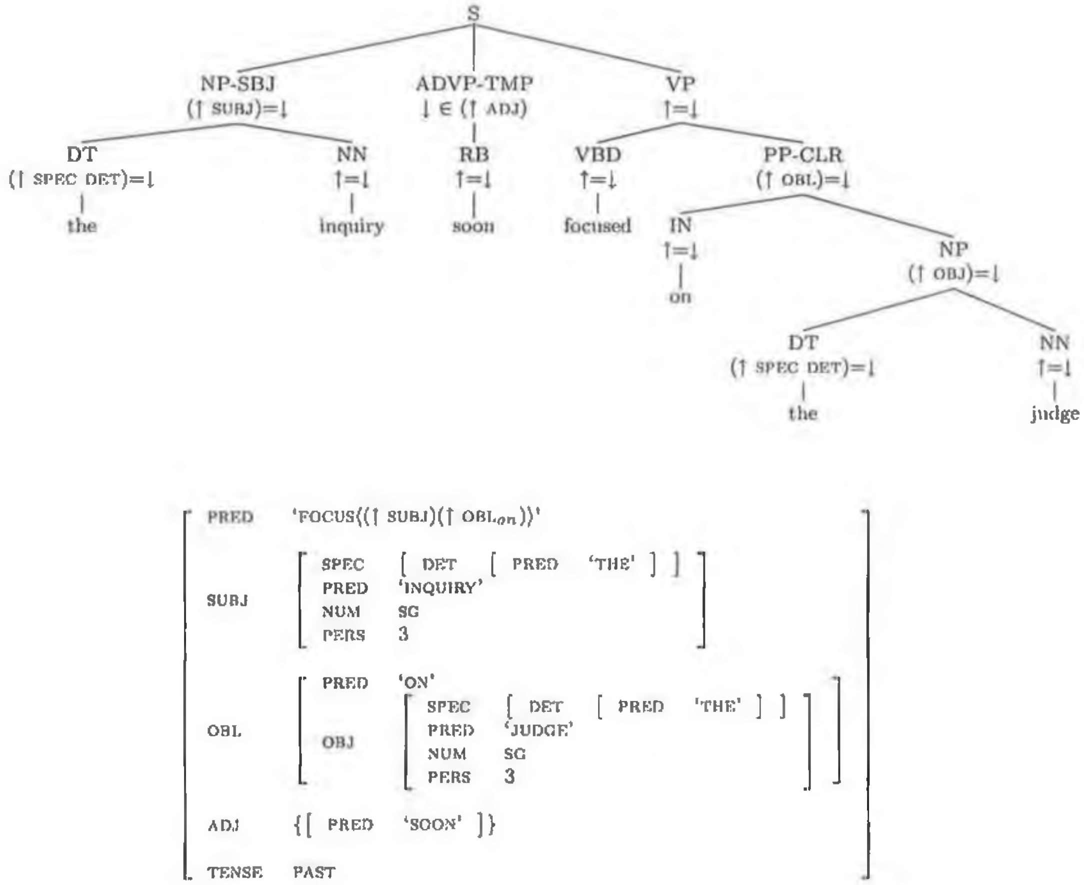


Figure 3.2: C- and F-Structure for Penn Treebank Sentence wsj_0267_72 '`The inquiry soon focused on the judge`'.

contains its own internal SUBJ:

`The judge thinks [`$_{\text{COMP}}$ `that the inquiry will resume ].`

XCOMP and XADJ are *open* functions not requiring an internal SUBJ. The subject is instead specified externally in the matrix phrase:

`The judge wants [`$_{\text{XCOMP}}$ `to open an inquiry ].`

While many linguistic theories state subcategorisation requirements in terms of phrase structure (CFG categories), Dalrymple (2001) questions the viability and universality of such an approach due to the variety of ways in which grammatical functions may be realised at the language-specific constituent structure level. LFG argues that subcategorisation requirements are best stated at the f-structure level, in functional rather than

Governable GFs	**Non-governable GFs**
SUBJ	ADJ
OBJ	XADJ
XCOMP	POSS
COMP	
OBJ_θ	
OBL_θ	

Table 3.1: Governable and Non-Governable Grammatical Functions in LFG

phrasal terms. This is due to the assumption that abstract grammatical functions are primitive concepts as opposed to derivatives of phrase structural position. In LFG, the subcategorisation requirements of a particular predicate are expressed by its semantic form: e.g. $\text{FOCUS}\langle(\uparrow \text{SUBJ})(\uparrow \text{OBL}_{on})\rangle$ in Figure 3.1.

The subcategorisation requirements expressed by semantic forms are enforced at f-structure level through *completeness* and *coherence* well-formedness conditions on f-structure (Kaplan and Bresnan, 1982):

> "An f-structure is *locally complete* iff it features all grammatical functions specified in the semantic form of its local PRED feature. An f-structure is *complete* iff it is locally complete and all its subsidiary f-structures are locally complete. An f-structure is *locally coherent* iff the only subcategorisable grammatical functions featured at that level are those listed in the semantic form value of the local PRED feature. An f-structure is *coherent* iff it is locally coherent and all its subsidiary f-structures are locally coherent."

Consider again the f-structure in Figure 3.2. The semantic form associated with the verb `focus` is $\text{FOCUS}\langle(\uparrow \text{SUBJ})(\uparrow \text{OBL}_{on})\rangle$. The f-structure is locally complete as it contains the SUBJ and an OBL with the preposition on specified by the semantic form. The f-structure also satisfies the coherence condition as it does not contain any governable grammatical functions other than the SUBJ and OBL required by the local PRED.

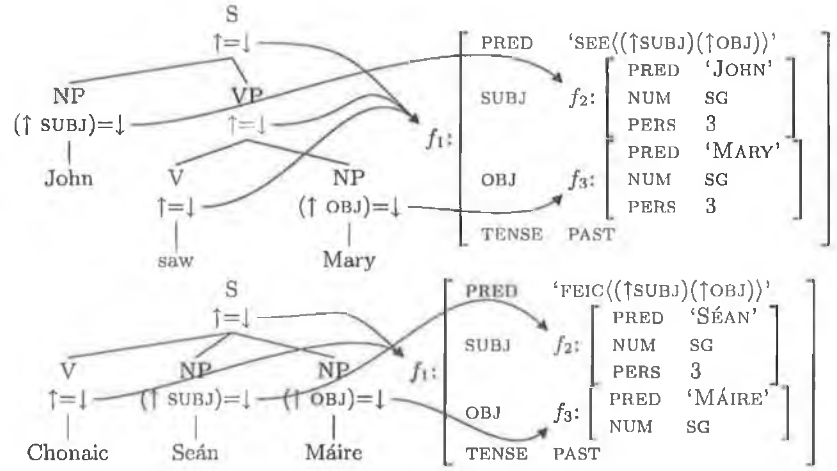

S
↑=↓
NP
(↑ SUBJ)=↓
John
VP
↑=↓
V
↑=↓
saw
NP
(↑ OBJ)=↓
Mary
f1:
PRED 'SEE⟨(↑SUBJ)(↑OBJ)⟩'
SUBJ f2: PRED 'JOHN' NUM SG PERS 3
OBJ f3: PRED 'MARY' NUM SG PERS 3
TENSE PAST
S
↑=↓
V
↑=↓
Chonaic
NP
(↑ SUBJ)=↓
Seán
NP
(↑ OBJ)=↓
Máire
f1:
PRED 'FEIC⟨(↑SUBJ)(↑OBJ)⟩'
SUBJ f2: PRED 'SÉAN' NUM SG PERS 3
OBJ f3: PRED 'MÁIRE' NUM SG
TENSE PAST

Figure 3.3: C- and F-structures for an English and Corresponding Irish Sentence

3.2.2 Motivation for LFG

LFG is particularly attractive for multilingual lexical extraction as the level of f-structure representation abstracts away from certain particulars of language-specific realisation (Butt et al., 1999, 2002). While languages differ with respect to surface representation, their f-structure representations may be the same or very similar. Figure 3.3 illustrates the point. Irish is typologically a VSO-language while English is an SVO-language. The same proposition expressed in Irish and English results in different c-structure configurations but in isomorphic (up to the values of PRED nodes) f-structures. An f-structure-based subcategorisation frame extraction methodology holds the advantage that it can remain agnostic about the underlying phrase-structure encodings and hence, in principle, it should be easier to port the extraction technology to different languages.

3.3 Automatic F-Structure Annotation

The first step in the application of our methodology is the production of a set of trees annotated with LFG f-structure information from which we generate f-structures. Given a set of trees the f-structure annotation algorithm of Cahill et al. (2002a); Burke et al. (2004b) traverses the trees, and automatically annotates the nodes in the trees with f-structure information. The trees can come from a treebank or from a probabilistic parser.

Here, I introduce the f-structure annotation algorithm with Penn-II treebank trees as the running example (in fact, the only difference between this and parser output is with respect to the Traces module).[2] The f-structure annotation algorithm has a modular architecture in order to facilitate extension or adaption of language specific details. The design is illustrated in Figure 3.4.

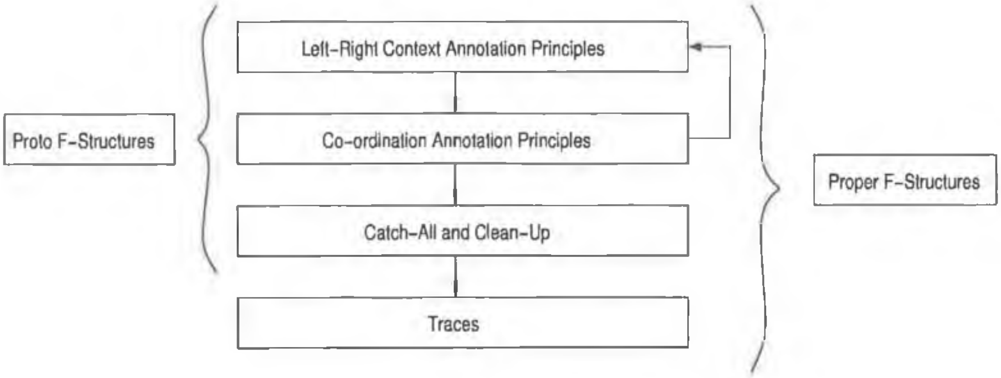

Left-Right Context Annotation Principles
Proto F-Structures
Co-ordination Annotation Principles
Proper F-Structures
Catch-All and Clean-Up
Traces

Figure 3.4: Outline of Algorithm to Generate Proto and Proper F-Structures.

The first three modules (Left-Right Context Annotation Principles, Co-ordination Annotation Principles and Catch-All and Clean-Up) are responsible for the production of 'proto' f-structures which capture basic but possibly incomplete predicate-argument structure. TOPIC, TOPICREL and FOCUS functions are used to represent 'moved' linguistic material due to long-distance dependency (LDD) phenomena such as topicalisation and wh-movement but are not resolved as arguments of the PREDS subcategorising for this material. The final module (Traces) resolves LDDs and produces 'proper' f-structures. Here, I will briefly outline each of the core components. For more details on automatic annotation, see (McCarthy, 2003) and (Burke, 2006).

3.3.1 Left-Right Context

The first step in the automatic annotation is the application of left-right context annotation principles to each local subtree of depth one (i.e. a CFG rule) which does not contain a coordinated structure. As previously mentioned, coordination is treated in a separate

[2] C.f. Section 3.3.4 and Chapter 5 for further explanation.

Left Context	Head	Right Context
DT: $\uparrow$ SPEC:DET=$\downarrow$ QP: $\uparrow$ SPEC:QUANT=$\downarrow$ JJ, ADJP: $\downarrow \in \uparrow$ ADJUNCT	NN, NNS ... $\uparrow=\downarrow$	NP: $\downarrow \in \uparrow$ APP PP: $\downarrow \in \uparrow$ ADJUNCT S, SBAR:$\uparrow$ RELMOD=$\downarrow$

Table 3.2: Simplified Sample NP Annotation Matrix

module. First the head-daughter of the local subtree is identified using an amended version of the head rules of Magerman (1995) and annotated with the LFG equation $\uparrow=\downarrow$. The identification and annotation of the rule head allow any additional nodes to be interpreted in terms of their structural relationship to the head: they either precede (left context) or follow it (right context). Annotation matrices encoding linguistic generalisations are consulted to assign f-structure annotations to these nodes based on their category label and rule context. To give a simple example, an NP occurring to the left of a VP head under an S is annotated as the sentential subject ($\uparrow$ SUBJ $=\downarrow$), while an NP to the right of the V head of a VP is assigned an object annotation ($\uparrow$ OBJ $=\downarrow$). Table 3.2 shows a simplified matrix for NP rules. The matrices are populated by hand using linguistic knowledge gleaned from treebank rules. Due to the Zipfian nature of the treebank grammars, it is sufficient to examine the subset of very frequently-occurring rule types to provide generalisations over the entire treebank. For each category, the set of most frequent rule types which together give minimum 85% coverage of all rule tokens are selected for manual examination. More detail on the annotation matrices can be found in (McCarthy, 2003) and (Burke, 2006).

3.3.2 Coordination

Penn-II analyses of coordinate structures are often relatively flat, thereby compounding the problem of their annotation. Constituents aside from the coordinated elements and the coordinating conjunction are annotated using the left-right context matrices described above. Figure 3.5 shows the annotation of a coordinated VP structure. The coordinating conjunction (CC) is assigned a head annotation and the VPs are annotated as conjuncts ($\downarrow\in\uparrow$COORD). The NP daughter receives the $\uparrow$OBJ=$\downarrow$ annotation based on the left right context rule for an NP occurring to the right of a VP's head daughter. Unlike-constituent coordination (UCP), which occurs very infrequently, is more complex to annotate automatically. Full details of how coordination is analysed and annotated are available in (McCarthy, 2003) and (Burke, 2006).

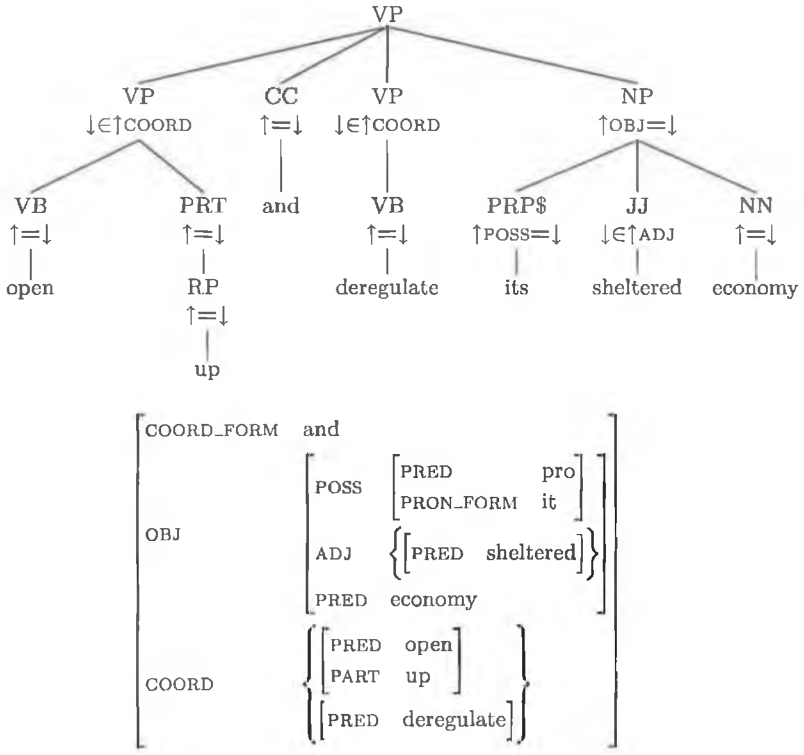


Figure 3.5: Annotating VP Coordination

3.3.3 Catch-All and Clean-Up

The purpose of this module is to correct or overwrite erroneous f-structure annotations and to assign default annotations to previously unannotated tree nodes. This is done in two ways. Penn-II functional labels (e.g. -TMP, -LOC etc.) are disregarded by the annotation matrices which generalise to category labels. In the catch-all and clean-up phase, default annotations are assigned to functionally labelled nodes, overwriting any existing annotation. For example, PP-CLR (-CLR indicates a close relationship to the verb) is always annotated as an oblique argument (↑OBL=↓) at this stage regardless of any annotation the PP may have been assigned by the left-right context rules.

It turned out to be necessary to parameterise this section of the annotation algorithm to produce f-structures more suited to the automatic extraction of semantic forms. There are a number of functional tags which the annotation algorithm did not exploit but which were necessary to improve the accuracy of extracted semantic forms. Too many S and SBAR nodes were being annotated as XCOMPs or COMPs by the original annotation algorithm when they were in fact adjuncts which was reflected in terms of lowered precision in the extracted lexical resources. To counter this problem, I now exploit the -PRP (purpose) and -ADV (adverbial) tags to overwrite any annotations on nodes with these tags to ↓∈↑ADJUNCT. I also encountered an issue which relates to inconsistency in the treebank annotation. Typically in the Penn-II treebank a verb particle is analysed using a PRT node with an RP daughter (the POS tag of the particle) (see Figure 3.6). However, there are cases in the treebank where a different analysis is used as shown in Figure 3.7. Here, the verb's particle (`down`) and its oblique argument (`to business`) are daughters of the ADVP-CLR node rather than sisters of the verb (`get`). I amended the annotation algorithm to adequately cope with the variance in treebank CFG tree representations.

The second task of the catch-all and clean-up phase is to correct any overgeneralisations made by the left-right context annotation principles. For example, the VP annotation matrix assigns an ↑OBJ=↓ to any NP occurring to the right of the head in a VP rule. However, if the verb has both a direct and an indirect object NP argument, we want to make this distinction. Annotating both with ↑OBJ=↓ would result in an irresolvable set of equations preventing the generation of an f-structure. The catch-all and clean-up rules correct the overgeneralisation and rewrite the annotation of the second NP (the indirect object) as ↑OBJ2=↓.

3.3.4 Trace Information

The Penn-II Treebank employs a rich arsenal of traces and empty productions (nodes which do not realise any lexical material) to co-index displaced material with the position where it should be interpreted semantically. In Figure 3.8, the indices on the S-TPC-1 and *T*-1 indicate that there is a correspondence between them: the topicalised S should be interpreted semantically at the location of the *T*-1 node as a COMP argument of **`said`**. The automatic annotation algorithm translates the traces into corresponding reentrancies in the f-structure representation by treating null constituents as full nodes. The

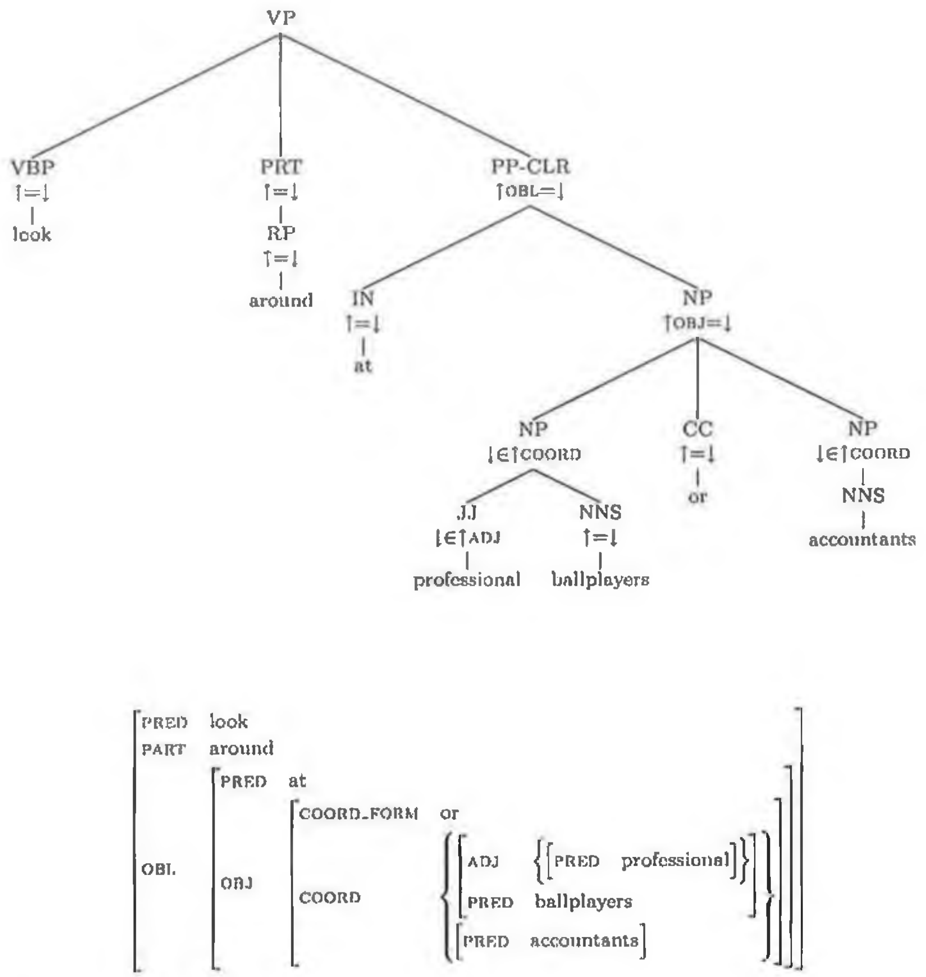


Figure 3.6: Penn-II Analysis of Verb-Particle Construction with PRT Node for the Sentence Fragment '`look around at professional ballplayers or accountants`'.

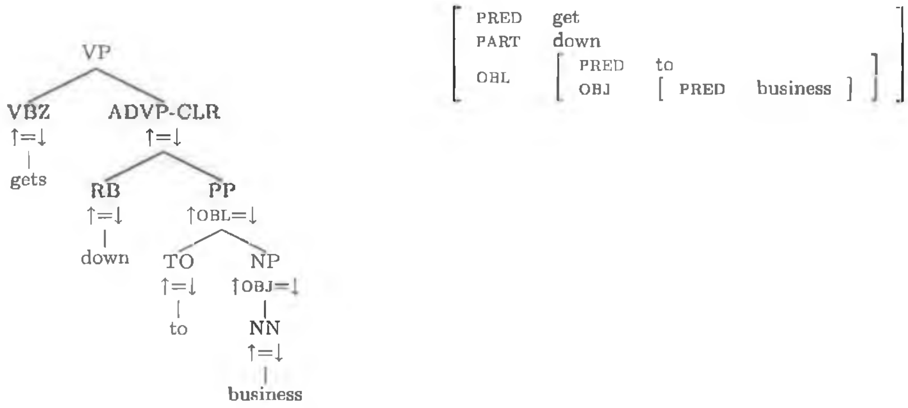


Figure 3.7: Penn-II Analysis of Verb-Particle Construction with ADVP-CLR Node for the Sentence Fragment '**get down to business**'.

link between the TOPIC and the COMP in the f-structure in Figure 3.8 is indicated by the index $\boxed{1}$. In contrast to Penn-II treebank trees, trees produced by parsing do not contain traces and empty nodes. For parser-based output, LDDs are resolved at f-structure rather than c-structure level, following theoretical LFG but with automatically extracted finite approximations of functional uncertainty equations (Cahill et al., 2004b). LDD resolution and its relationship to lexical extraction are discussed in Chapters 4 and 5.

3.3.5 From Trees to F-Structures

Once the trees have been successfully annotated with f-structure information, the equations are collected from each node and converted into PROLOG format. The set of equations is then passed to a PROLOG constraint solver, which is based on and extends the constraint solver of (Gazdar and Mellish, 1989).

3.4 Lexical Extraction

If our automatically produced f-structures are of high quality, we can reverse engineer semantic forms. Our semantic form extraction methodology is based on and substantially extends the granularity and coverage of an idea proposed by (van Genabith et al., 1999a):

> "For each f-structure generated, for each level of embedding we determine the local PRED value and collect the subcategorisable grammatical functions

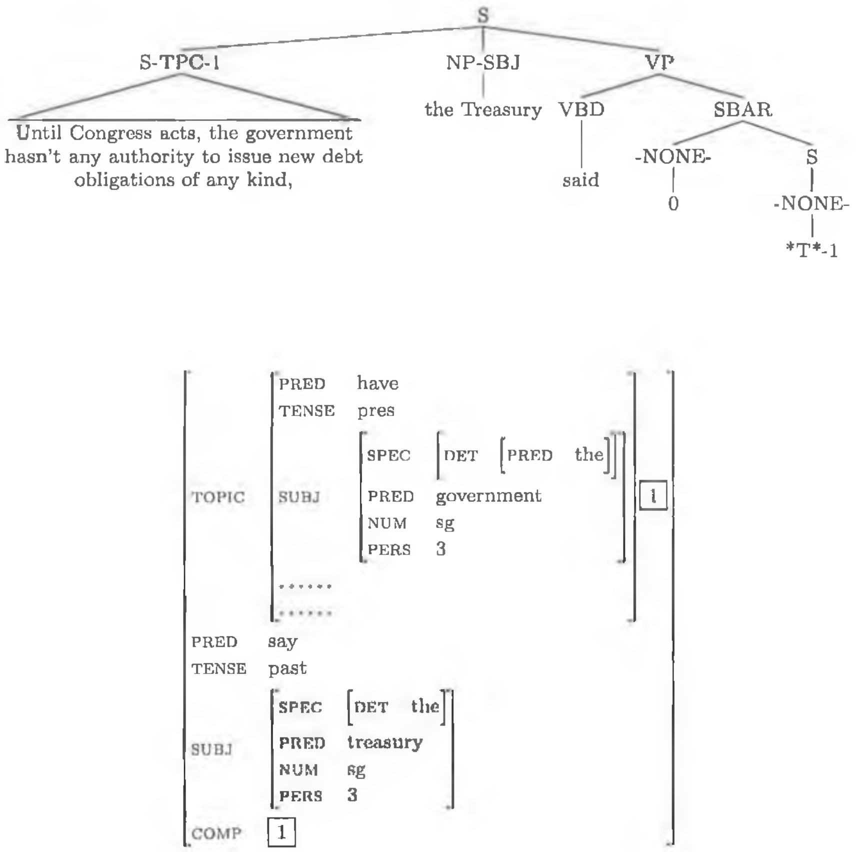


Figure 3.8: Use of Reentrancy between TOPIC and COMP to capture a Long-Distance Dependency in wsj_0008_2 “**Until Congress acts , the government hasn’t any authority to issue new debt obligations of any kind, the Treasury said .**”

present at that level of embedding.”

To illustrate the methodology, consider the automatically generated f-structure in Figure 3.9 for tree wsj_0003_22 in the Penn-II Treebank. It is crucial to note that in the automatically generated f-structure the values of the `pred` features are lemmas or lexical base forms (`impose`, `in`, `of` and `on`) and not yet semantic forms — i.e. a lemma followed by a frame type (argument list) — as in theoretical LFG (Section 3.2.1). Applying the method described above to the f-structure in Figure 3.9, we recursively extract the following non-empty semantic forms: `impose([subj,obj,obl:on])`, `in([obj])`, `of([obj])` and `on([obj])`. In effect, semantic forms are reverse engineered from automatically generated f-structures for treebank or parse trees.

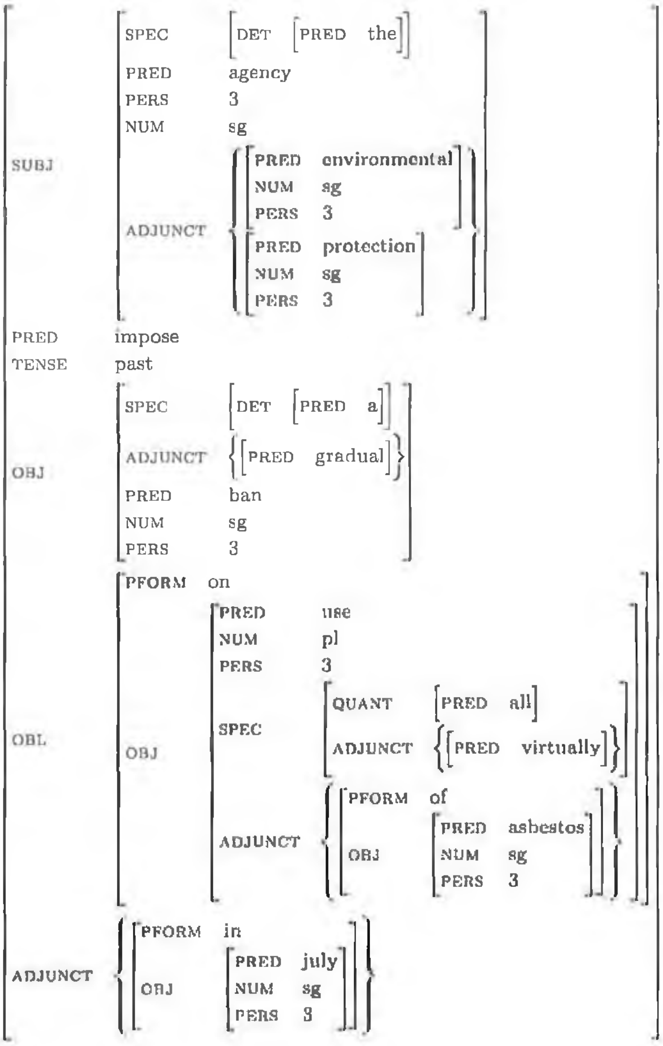
The automatically induced semantic forms contain the following subcategorisable syntactic functions:

SUBJ	OBJ	OBJ2	OBL$_{prep}$	OBL2$_{prep}$	COMP	XCOMP	PART$_{part}$

PART is not a syntactic function in the strict sense but we decided to capture the relevant co-occurrence patterns of verbs and particles in the semantic forms. Just as OBL$_{prep}$ includes the prepositional head of the PP, PART$_{part}$ includes the actual particle which occurs e.g. `add([subj,obj,part:up])`.

In the work presented here I substantially extend and scale the approach of (van Genabith et al., 1999a) as regards coverage and granularity. First, as I show in Chapters 4 and 5, I scale the original approach to the Penn-III treebank and the British National Corpus (BNC). The work described in van Genabith et al. (1999a) was proof of concept on 100 trees. Second, in contrast to van Genabith et al. (1999a) (and many other approaches), the approach fully reflects LDDs, indicated in terms of traces and corresponding f-structure re-entrancies for the Penn-III Treebank, or resolved at f-structure level for parser-produced trees. Third, in addition to abstract syntactic function-based subcategorisation frames I also compute frames for syntactic function-CFG category pairs, both for the verbal heads and their arguments and also generate pure CFG-based subcategorisation frames. Fourth, in contrast to van Genabith et al. (1999a) (and many other approaches) the method differentiates between frames for active or passive constructions. Fifth, in contrast to van Genabith et al. (1999a), the method associates conditional probabilities with frames. Finally, we successfully port the methodology to Spanish, German and Chinese (Chapter 6).

Below I expand on the following aspects of the system: capture of categorial information, capture of long distance dependency information, distinction between active and passive frames, classification of clausal complements, treatment of coordination and, finally, computation of conditional probabilities.



impose([subj,obj,obl:on])
in([obj])
of([obj])
on([obj])

Figure 3.9: Automatically Generated F-Structure and Extracted Semantic Forms for the Penn-II Treebank String wsj_0003_22 "`In July, the Environmental Protection Agency imposed a gradual ban on virtually all uses of asbestos.`"

3.4.1 Categorial Information

In order to capture CFG-based categorial information, the annotation algorithm adds an optional CAT feature to the f-structures automatically generated from the treebank or parser-generated trees. Its value is the syntactic category of the lexical item whose lemma gives rise to the `pred` value at that particular level of embedding. This makes it possible to classify words and their semantic forms based on their syntactic category and

Conflated Category	Penn Treebank Category
JJ	JJ
	JJR
	JJS
N	NN
	NNS
	NNP
	NNPS
	PRP
RB	RB
	RBR
	RBS
V	VB
	VBD
	VBG
	VBN
	VBP
	VBZ
	MD

Table 3.3: Conflation of Penn Treebank Tags

reduces the risk of inaccurate assignment of subcategorisation frame frequencies due to POS ambiguity, distinguishing for example between the nominal and verbal occurrences of the lemma `fight`. With this extension, the output for the verb `impose` in Figure 3.9 is `impose(v,[subj,obj,obl:on])`. For some of our experiments, I conflate the different verbal (and other) tags used in the Penn Treebanks to a single verbal marker (Table 3.3). As a further extension, the extraction procedure reads off the syntactic category of the head of each of the subcategorised syntactic functions: `impose(v,[subj(n),obj(n),obl:on])`.[3] In this way, the methodology is able to produce surface syntactic (`impose(v,[n,n,pp])`) as well as abstract functional subcategorisation details. Dalrymple (2001) argues that there are cases, although exceptional, where constraints on syntactic category are an issue in subcategorisation. Our system can provide this information as well as details of grammatical function.

[3] We do not associate a syntactic category with OBLs as they are always PPs.

3.4.2 Long Distance Dependencies

In contrast to many other approaches to lexical extraction, our methodology produces lexical entries reflecting the LDDs in the source data structures. Consider again the f-structure in Figure 3.8. As the TOPIC has correctly been resolved as the COMP of the local PRED `say`, I correctly extract the frame `say([subj,comp])` as opposed to incorrectly associating `say` with an intransitive frame (`say([subj])`).

3.4.3 Passive

I extend the basic extraction algorithm is to deal with passive voice and its effect on subcategorisation behaviour. Consider Figure 3.10: not taking into account that the example sentence is a passive construction, the extraction algorithm extracts `outlaw([subj])`. This is incorrect as `outlaw` is a transitive verb and therefore requires both a subject and an object to form a grammatical sentence in the active voice. To cope with this problem, the extraction algorithm uses the feature value pair `passive:+`, which appears in the f-structure at the level of embedding of the verb in question, to mark that predicate as occurring in the passive: `outlaw([subj],p)`.

3.4.4 Clausal Complements

The syntactic functions COMP and XCOMP refer to clausal complements with different predicate control patterns as described in Section 3.2.1. However, as it stands neither of these functions betray anything about the syntactic nature of the constructs in question. Many lexicons, both automatically acquired and manually created, are more fine-grained in their approaches to subcategorised clausal arguments, differentiating for example between a *that*-clause and a *to+infinitive* clause (Ushioda et al., 1993). With only a slight modification, the system, along with the details provided by the automatically generated f-structures, allows the extraction of frames with an equivalent level of detail. For example, to identify a *that*-clause, I use the feature-value pair `that:+` at f-structure level to read off the following subcategorisation frame for the verb `add`: `add([subj,comp(that)])`. Using the feature-value pair `to_inf:+`, I identify *to+infinitive* clauses, resulting in the following frame for the verb `want`: `want([subj,xcomp(to_inf)])`. The methodology can also de-

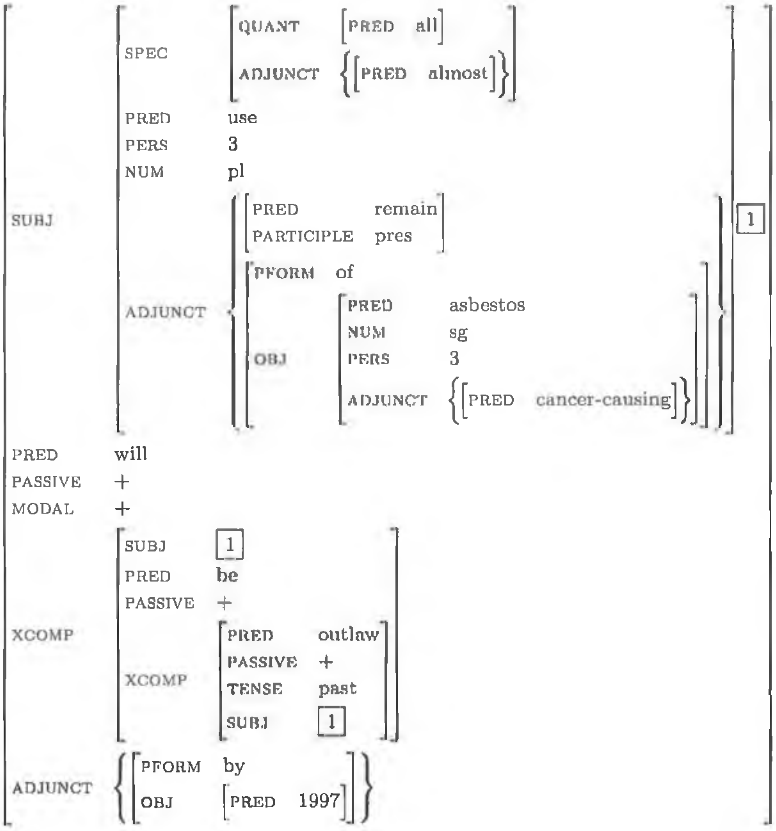


Figure 3.10: Automatically Generated F-Structure for the Penn-II Treebank string wsj_0003_23 "`By 1997, almost all remaining uses of cancer-causing asbestos will be outlawed.`"

rive control information about open complements. In Figure 3.10, the re-entrant `xcomp` subject is identical to the subject of `will` in the matrix clause which allows the induction of information about the nature of the external control of the `xcomp` (i.e. whether it is subject or object control).

3.4.5 Coordination

The basic extraction algorithm searches each f-structure in turn to locate the PRED and its subcategorisable grammatical functions. In the case of coordination, however, the PREDs occur within the conjuncts which make up the value of the COORD feature as illustrated in Figure 3.11. The object of the PRED values `harass` and `punish` is not contained in their local f-structure but in that of the COORD feature. When the extraction algorithm

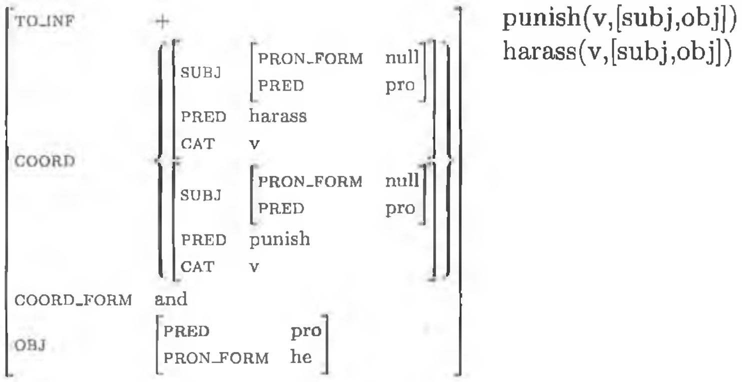


Figure 3.11: Example of Coordination and Extracted Verbal Semantic Forms from Sentence wsj_0049_42: `Cartoonist Garry Trudeau is suing The Writers' Guild of America for 11 million, alleging it mounted a campaign to harass and punish him for crossing a screenwriters' picket line.`

encounters an f-structure without a PRED value, with a COORD value and containing one or more subcategorisable grammatical functions, these grammatical functions (in this case OBJ) are stored until the coordinated PRED values are found and then they are merged with the list of locally occurring grammatical functions (in this case SUBJ) for each PRED.

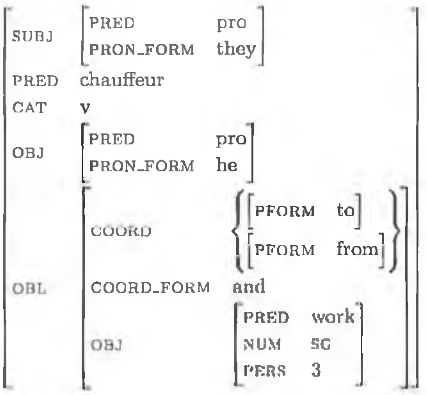
Coordination also complicates the assignment of a specific preposition to an oblique function. Normally the extraction algorithm looks for the PFORM feature contained in the OBL value. When a sentence contains coordinated OBLs as in Figure 3.12, the algorithm must look into the conjunct set to find the preposition. The verb must in addition be assigned two rather than one subcategorisation frame, one for each preposition.

3.4.6 Conditional Probabilities

In order to estimate the likelihood of the co-occurrence of a predicate with a particular argument list, I compute conditional probabilities for subcategorisation frames based on the number of token occurrences in the corpus:

$$\mathcal{P}(ArgList|\Pi) = \frac{count(\Pi\langle ArgList\rangle)}{\sum_{i=1}^{n} count(\Pi\langle ArgList_i\rangle)}$$

where $ArgList_1 \ldots ArgList_n$ are the possible argument lists which can occur for Π. Due to variations in verbal subcategorisation across domains, probabilities are also useful for



chauffeur(v,[subj,obj,obl:to])
chauffeur(v,[subj,obj,obl:from])

Figure 3.12: Example of Coordinated Obliques and Extracted Verbal Semantic Forms from Sentence wsj_0267_13: `Bailiffs claimed they were required to chauffeur him to and from work, mow his lawn, chop his wood, fix his car and even drop by his house to feed his two grown mutts, Dixie and Husky.`

predicting the way in which verbs behave in certain contexts. Furthermore, I use the conditional probabilities to filter possible error judgements by the system.

To demonstrate the output of our system, Tables 3.4, 3.5 and 3.6 show, with varying levels of analyses, the attested semantic forms for the verb `accept` with their associated conditional probabilities from the Penn-II Treebank. The effect of differentiating between the active and passive occurrences of verbs can be seen in the different conditional probabilities associated with the intransitive frame ([subj]) of the verb `accept` in Table 3.4 and Table 3.5.[4] Table 3.6 shows the joint grammatical function/syntactic category-based subcategorisation frames.

3.5 Variations of the Basic Architecture

Figure 3.13 gives an overview of the architecture presented in this chapter. A set of trees is automatically annotated with LFG f-structure information. The f-structure equations from each tree are collected and passed to the constraint solver which produces a corresponding set of f-structures. Lexical information is then extracted from the f-

[4] Given these, it is possible to condition frames on both lemma (Π) and voice (v: active/passive):

$$\mathcal{P}(ArgList|\Pi, v) = \frac{count(\Pi\langle ArgList, v\rangle)}{\sum_{i=1}^{n} count(\Pi\langle ArgList_i, v\rangle)}$$

Semantic Form	Occurrences	Conditional Probability
accept([subj,obj])	122	0.813
- accept([subj])	11	**0.073**
accept([subj,comp])	5	0.033
accept([subj,obl:as])	3	0.020
accept([subj,obj,obl:as])	3	0.020
accept([subj,obj,obl:from])	3	0.020
accept([subj,obj,obl:at])	1	0.007
accept([subj,obj,obl:for])	1	0.007
accept([subj,obj,xcomp])	1	0.007

Table 3.4: Semantic Forms for the Verb accept

Semantic Form	Occurrences	Conditional Probability
accept([subj,obj])	122	0.813
- accept([subj],p)	**9**	**0.060**
accept([subj,comp])	5	0.033
accept([subj,obl:as],p)	3	0.020
accept([subj,obj,obl:as])	3	0.020
accept([subj,obj,obl:from])	3	0.020
- accept([subj])	**2**	**0.013**
accept([subj,obj,obl:at])	1	0.007
accept([subj,obj,obl:for])	1	0.007
accept([subj,obj,xcomp])	1	0.007

Table 3.5: Semantic Forms for the Verb accept Mrked with p for Passive Use.

Semantic Form	Occurrences	Conditional Probability
accept([subj(n),obj(n)])	116	0.773
accept([subj(n)])	11	0.073
accept([subj(n),comp(that)])	4	0.027
accept([subj(n),obj(n),obl:from])	3	0.020
accept([subj(n),obl:as])	3	0.020
Other	13	0.087

Table 3.6: Semantic Forms for the Verb accept including Syntactic Category for each Grammatical Function.

structures. The modularity of the architecture supports three lexical extraction variations: treebank-based lexical extraction, parser-based lexical extraction and multilingual lexical extraction.

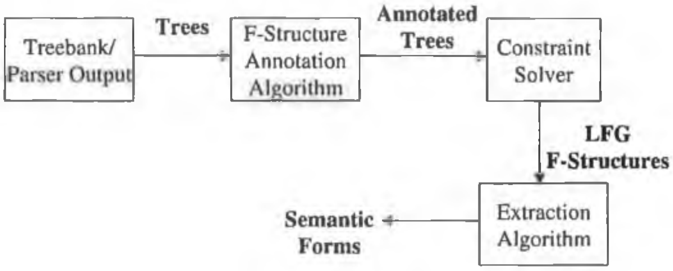


Figure 3.13: Lexical Extraction System Architecture

3.5.1 Treebank-based Lexical Extraction

Using the 75,000 trees from the Penn-III Treebank as a starting point, we automatically produce a set of f-structures as described in Section 3.3. From this set of f-structures I automatically induce an LFG-based lexicon. Details of this extraction as well as coverage and quality of the induced lexical resources are presented in Chapter 4.

3.5.2 Parser-based Lexical Extraction

Using statistical parsers (Collins, 1999; Charniak, 2000; Bikel, 2002) trained on the Penn-II Treebank, we parse the written text component of the BNC. The resulting trees are then automatically annotated and lexical resources are induced from the automatically produced f-structures. A full description of this work including qualitative and quantitative evaluations of the induced resources is presented in Chapter 5.

3.5.3 Multilingual Lexical Extraction

Given a treebank or parsing resources for languages other than English, the subcategorisation extraction methodology presented in this chapter carries over unchanged to multilingual subcategorisation frame extraction. In Chapter 6, I present basic results for treebank-based approaches to Spanish, German and Chinese, three typologically markedly different languages. In our experiments we use the Cast3LB, TIGER and Penn-CTB treebanks (Figure 3.14).

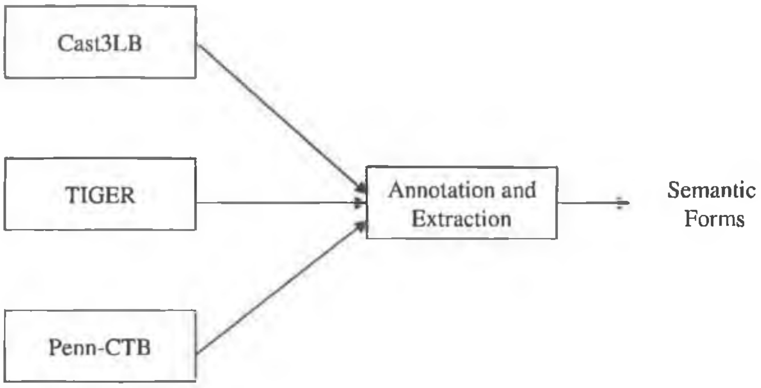


Figure 3.14: Overview of Multilingual Lexical Extraction

3.6 Summary

In this chapter I presented a methodology for the automatic extraction of lexical resources based on an automatic LFG f-structure annotation algorithm. I first outlined Lexical Functional Grammar, the theoretical basis for the methodology. A prerequisite for the automatic induction of lexical resources is the production of an f-structure-annotated set of trees. I described how such a resource can be produced automatically using the annotation algorithm of (Cahill et al., 2002a; Burke et al., 2004b). Lexical resources are reverse engineered from automatically produced f-structures. The extracted subcategorisation frames express subcategorisation requirements in terms of grammatical functions following theoretical LFG or, optionally, in terms of syntactic category. Long distance dependencies in the source data structures are reflected in the induced lexical entries. The system differentiates between active and passive frames and associates specific prepositions and particles with the relevant frames. In addition, probabilities are assigned to frames conditional on the lemma, facilitating the optimisation of the extracted lexicon for quality or coverage through statistical filtering. The modularity of the extraction architecture facilitates three different implementations: treebank-based lexical extraction, parser-based lexical extraction and multilingual lexical extraction. In subsequent chapters I describe these implementations in detail, demonstrating the quality and robustness of the methodology.

Chapter 4

Treebank-Based Semantic Form Extraction from Penn-II and Penn-III

4.1 Introduction

In Chapter 3, I outlined a methodology for the automatic induction of lexical resources. This chapter describes the application of the methodology to the Penn-III Treebank, 75,000 manually produced trees from a variety of text genres. The trees are first automatically annotated with LFG f-structure information to produce a corresponding set of f-structures from which I induce or reverse-engineer a large lexical resource. As the lexical resources are integrated into a parsing system (Cahill et al., 2004b), their quality is important. I describe extensive evaluation experiments carried out to assess the quality of the extracted resources.

Various approaches have been taken to the qualitative evaluation of lexical extraction systems. As one of the priorities of the evaluation is to demonstrate the scalability of the approach, I carry out extensive evaluations (over 3,000 verbs) against two existing hand-crafted lexical resources: COMLEX (Grishman et al., 1994; Macleod et al., 1994) and the OALD (Hornby, 1980). The advantages and disadvantages of using such resources as gold standards are also discussed here.

The extracted lexical resources are required for long distance dependency (LDD) resolution in the parsing system of Cahill et al. (2004b). Following (Cahill et al., 2004b), I carry out a task-based evaluation measuring the improvements in parser performance achieved by incorporating the induced lexical information. I demonstrate the coverage of the lexical extraction system on unseen text and the rate at which new lexical information is induced.

The chapter begins with a brief description of the Penn-III treebank and an evaluation of the automatic annotation step. Commonly used evaluation techniques for induced lexical resources are discussed in Section 4.3. Section 4.4 describes the evaluation carried out against COMLEX. Section 4.5 describes the experiments which use the OALD as a gold standard. In Section 4.6, I describe how the lexical resources are employed at parse time and their effect on the quality of parser output. Finally in Section 4.7, I provide a quantitative evaluation demonstrating the coverage of the induced resources on new text, the rate of accession of new text and issues of lexical domain specificity associated with the source data. Part of the work reported in this chapter has been published in O'Donovan et al. (2005a) and Cahill et al. (2004b).

4.2 Automatically Annotating the Penn-II and Penn-III Treebanks

Most of the early work on automatic f-structure annotation (e.g. (van Genabith et al., 1999b; Sadler et al., 2000; Frank et al., 2003)) was applied only to small data sets (less than 200 sentences) and was largely 'proof of concept'. However, more recent work (Cahill et al., 2002a; Burke et al., 2004b) as described in Chapter 3 has evolved and scaled up annotation techniques to the Penn-II Treebank, containing more than 1,000,000 words and 50,000 sentences. For the purposes of lexical extraction, I also wished to annotate and extract semantic forms from Penn-III. Aside from the 50,000 parsed WSJ strings of Penn-II, Penn-III contains parses for a subsection of the Brown Corpus (almost 385,000 words in 24,000 trees) taken from the following text genres: popular lore; belles lettres, biography, memoires, etc; general fiction; mystery and detective fiction; science fiction;

adventure and western fiction; romance and love story; humour. The annotation scheme is the same as that for the WSJ section (of Penn-II and Penn-II) with one exception. The -CLR tag used in Penn-II and exploited by the automatic annotation algorithm in the identification of obliques is no longer used. I therefore amend the way in which the algorithm makes the adjunct/oblique distinction. For Penn-III the algorithm annotates all PPs which do not carry a Penn adverbial functional tag (such as -TMP or -LOC) and which occur as the sisters of the verbal head of a VP as obliques. In addition, the algorithm annotates as obliques PPs associated with -PUT (locative complements of the verb put) or -DTV (second object in ditransitives) tags.[1]

4.2.1 Evaluation of the Automatic Annotation Algorithm

	DCU 105	**PARC 700**
Precision	96.52%	87.95%
Recall	96.62%	86.21%
F-Score	96.57%	87.07%

Table 4.1: Results of F-Structure Evaluation

In order to ensure the quality of the semantic forms extracted by our method, I must first ensure the quality of the f-structure annotations. The results of two different evaluations of the automatically generated f-structures are presented in Table 4.1. Both use the evaluation software and triples encoding presented in (Crouch et al., 2002). The first of these is against the DCU 105, a gold standard set of 105 hand-coded f-structures from Section 23 of the Penn Treebank as described in (Burke et al., 2004b). For the full set of annotations we achieve Precision of over 96.5% and Recall of over 96.6%. There is, however, a risk of overfitting when evaluation is limited to a gold standard of this size. More recently, Burke et al. (2004a) carried out an evaluation of the automatic annotation algorithm against the publicly available PARC 700 Dependency Bank (King et al., 2003), a set of 700 randomly selected sentences from Section 23 which have been parsed, converted to dependency format and manually corrected and extended by human validators. The resulting precision is almost 88% and Recall is over 86% (Table 4.1). The PARC

[1] There is a risk of over generation but I hope to filter out incorrect frames using frequency thresholds.

700 Dependency Bank differs substantially from both the DCU 105 f-structure bank and the automatically generated f-structures as regards the style of linguistic analysis, feature nomenclature and feature geometry. Some, but not all, of these differences are captured by automatic conversion software. A detailed discussion of the issues inherent to this process and a full analysis of results is presented in (Burke et al., 2004a). Results broken down by grammatical function for the PARC 700 evaluation are presented in Table 4.2.

4.2.2 Scale of Extracted Lexical Resource

I extract semantic forms for 4362 verb lemmas from Penn-III (WSJ and Brown combined). Table 4.3 shows the number of distinct verbal semantic form types (i.e. lemma and frame type combination) extracted. Discriminating obliques by associated preposition and recording particle information, the algorithm finds a total of 21,005 semantic form types, 16,000 occurring in active voice and 5,005 in passive voice. When the obliques are parameterised for prepositions and particles included for particle verbs, I find an average of 4.82 semantic form types per verb. Without the inclusion of details for individual prepositions or particles, there was an average of 3.45 semantic form types per verb (Table 4.4). Table 4.5 shows the numbers of distinct frame types extracted from Penn-III, ignoring PRED values.[2] I give two columns of figures, one where all oblique arguments are condensed into one OBL function and all particle arguments are condensed into PART, and the other where I differentiate between `obl:to` (e.g `give`), `obl:on` (e.g. `rely`), `obl:for` (e.g. `compensate`) etc. and likewise for particles. Collapsing obliques and particles into simple functions I extract 50 frame types. Discriminating particles and obliques by preposition I extract 1084 frame types. I also show the result of applying absolute thresholding techniques to the semantic forms induced. Applying an absolute threshold of 5 occurrences, I still generate 221 frame types from the combined Penn-III. Briscoe and Carroll (1997), by comparison, employ 163 distinct predefined frames.

[2] To recap, if two verbs have the same subcategorisation requirements (e.g. `give([subj,obj,obj2])`, `send([subj,obj,obj2])`), then that frame `[subj,obj,obj2]` is counted only once.

	Precision	Recall	F-Score
ADEGREE	1038/1218 = 85	1038/1290 = 80	83
ADJUNCT	2295/3046 = 75	2295/2995 = 77	76
AQUANT	9/10 = 90	9/13 = 69	78
COMP	212/236 = 90	212/257 = 82	86
CONJ	452/534 = 85	452/552 = 82	83
COORD_FORM	227/307 = 74	227/252 = 90	81
DET_FORM	948/962 = 99	948/964 = 98	98
FOCUS_INT	0/0 = 0	0/5 = 0	0
MOD	411/483 = 85	411/573 = 72	78
NUM	3721/3980 = 93	3721/4145 = 90	92
NUMBER	259/295 = 88	259/297 = 87	88
NUMBER_TYPE	410/424 = 97	410/440 = 93	95
OBJ	1675/1800 = 93	1675/1866 = 90	91
OBJ_THETA	5/12 = 42	5/11 = 45	43
OBL	127/236 = 54	127/173 = 73	62
OBL_AG	38/43 = 88	38/45 = 84	86
OBL_COMPAR	5/10 = 50	5/15 = 33	40
PASSIVE	186/208 = 89	186/238 = 78	83
PCASE	40/43 = 93	40/52 = 77	84
PERF	79/88 = 90	79/86 = 92	91
POSS	186/200 = 93	186/205 = 91	92
PRECOORD_FORM	0/0 = 0	0/6 = 0	0
PROG	169/174 = 97	169/203 = 83	90
PRON_FORM	507/547 = 93	507/531 = 95	94
PRON_INT	0/0 = 0	0/6 = 0	0
PRON_REL	103/145 = 71	103/119 = 87	78
PROPER	625/761 = 82	625/744 = 84	83
PRT_FORM	32/39 = 82	32/46 = 70	75
QUANT	55/69 = 80	55/71 = 77	79
STMT_TYPE	962/1066 = 90	962/1094 = 88	89
SUBJ	1578/1718 = 92	1578/1779 = 89	90
SUBORD_FORM	157/190 = 83	157/195 = 81	82
TENSE	1002/1022 = 98	1002/1051 = 95	97
TOPIC_REL	105/174 = 60	105/119 = 88	72
XCOMP	414/462 = 90	414/478 = 87	88

Table 4.2: Results by Feature Name of Evaluation against the PARC 700

	Without Prep/Part	With Prep/Part
Sem. Form Types	15166	21005
Active	11038	16000
Passive	4128	5005

Table 4.3: Number of Semantic Form Types for Penn-III (WSJ and Brown)

	Without Prep/Part	With Prep/Part
Avg. Sem. Form Types	3.45	4.82

Table 4.4: Average Number of Semantic Form Types per Verb for Penn-III (WSJ and Brown)

4.3 Approaches to Lexical Resource Evaluation

Most of the previous approaches to the automatic acquisition of lexical resources discussed in Chapter 2 have been evaluated to different degrees. In general, a small number of frequently occurring verbs are selected and the subcategorisation frames extracted for these verbs (from some quantity of unseen test data) are compared to a gold standard. The gold standard is either manually custom-made based on the test data or adapted from an existing external resource such as the OALD (Hornby, 1980) or COMLEX (Macleod et al., 1994). There are advantages and disadvantages to both types of gold standard. While it is time-consuming to manually construct a custom-made standard, it is a fairer measure of extraction system performance as it only contains the subcategorisation frames exhibited in the test data. In addition, it enables the evaluation of the relative frequencies associated with the extracted frames. Using an existing externally produced resource is quicker but the gold standard may contain many more frames than those which occur in the data from which the test lexicon is induced or, indeed, may omit relevant correct frames contained in the data. As a result, systems generally score better against custom-made, manually established gold standards. Carroll and Rooth (1998) achieve an f-score of 77% against the OALD when they evaluate a selection of 100 verbs with absolute frequency of greater than 500 each. Their system recognises 15 frames and these do not contain details of subcategorised-for prepositions. Still, to date this is the largest number of verbs used in any of the evaluations of the systems for English described in Chapter 2. Sarkar

	Without Prep/Part	With Prep/Part
# Frame Types	50	1084
# Singletons	6	544
# Twice Occurring	2	147
# Occurring max. 5	12	863
# Occurring > 5	38	221

Table 4.5: Number of Frame Types for Verbs for Penn-III (WSJ and Brown)

and Zeman (2000) evaluate 914 Czech verbs against a custom-made gold standard and record a token recall of 88%. However, their evaluation does not examine the extracted subcategorisation frames but rather the argument-adjunct distinctions posited by their system. The largest lexical evaluation we know of is that of Schulte im Walde (2002b) for German. She evaluates 3000 German verbs with a token frequency of between 10 and 2000 against the Duden (Dudenredaktion, 2001). I will refer to this work and the methods and results presented by Schulte im Walde in more detail in Section 4.4.

Korhonen (2002) points out that the real demonstration of the quality of a lexical resource is the improvement the resources effect in an application task. Briscoe and Carroll (1997) incorporate subcategorisation frame frequency information into a non-lexicalised statistical parser. On a test set of 250 sentences from the SUSANNE treebank, bracket precision and recall were unaffected while the no crossing bracket score increased by 7%. This improvement, however, was not found to be statistically significant. In a larger experiment employing a grammatical relation-based evaluation, Carroll et al. (1998b) report a statistically significant improvement in precision of 9%. The formalism-specific treebank-based lexical extraction models discussed in Chapter 2 (Hockenmaier, 2003; Miyao et al., 2003) are evaluated purely on their performance in a parsing task rather than against a gold standard. As these formalisms are highly lexicalised with an (almost) invariant rule set, parser performance is a clear indicator of lexicon quality. To evaluate, Hockenmaier (2003) and Miyao et al. (2003) use an automatically annotated version of Section 23 of the WSJ as a gold standard. They then parse the strings from Section 23 and compare their output to the automatically annotated gold standard output.

4.4 COMLEX Evaluation

I carried out a large-scale evaluation of the automatically induced lexicon (2,847 active verb lemmas for Penn-II and 3,535 for Penn-III, as well as 2430 passive verb lemmas from Penn-III) against the COMLEX resource. To our knowledge this is the most extensive evaluation ever carried out for English lexical extraction. Finding a common format for the gold standard and induced lexical entries is a non-trivial task. To ensure that I did not bias the evaluation in favour of either resource, I carried out two different mappings for the frames from Penn-II and Penn-III: COMLEX-LFG Mapping I and COMLEX-LFG Mapping II. For each mapping I carried out six basic experiments (and two additional ones for COMLEX-LFG Mapping II) for the active subcategorisation frames extracted. Within each experiment, the following factors were varied: level of prepositional phrase detail, level of particle detail, *relative* threshold (1% or 5%) and incorporation of an expanded set of directional prepositions. Using the second mapping I also evaluated the automatically extracted passive frames and experimented with absolute thresholds. Direct comparison of subcategorisation frame acquisition systems is difficult due to variations in the number of frames extracted, the number of test verbs, the gold standards used, the size of the test data and the level of detail in the subcategorisation frames (e.g. whether they are parameterised for specific prepositions or not). Therefore, in order to establish a baseline against which to compare our results, following (Schulte im Walde, 2002b), I assigned the two most frequent frame types (transitive and intransitive) by default to each verb and compared this 'artificial' lexicon to the gold standard. The section concludes with a full discussion of the reported results

4.4.1 COMLEX

I evaluate the induced semantic forms against COMLEX (Macleod et al., 1994), a computational machine-readable lexicon containing syntactic information for approximately 38,000 English headwords. Its creators paid particular attention to the encoding of more detailed subcategorisation information than either the OALD or the LDOCE (Proctor, 1978), both for verbs and for nouns and adjectives which take complements (Grishman et al., 1994). By choosing to evaluate against COMLEX, we set our sights high: the extracted semantic forms are fine-grained and COMLEX is considerably more detailed than the OALD or LDOCE. While the system can generate semantic forms for any lemma (regardless of part of speech) which induces a PRED value, I have thus far evaluated the automatic generation of subcategorisation frames for verbs only. COMLEX defines 138 distinct verb frame types without the inclusion of specific prepositions or particles.

As COMLEX contains information other than subcategorisation details, it was necessary to extract the subcategorisation frames associated with each verbal lexicon entry. The following is a sample entry for the verb **reimburse**:

```
(VERB  :ORTH "reimburse"  :SUBC  ((NP-NP)
                                  (NP-PP :PVAL ("for"))
                                  (NP)))
```

Each entry is organised as a nested set of typed feature-value lists. The first symbol (i.e. VERB) gives the part of speech. The value of the :ORTH feature is the base form of the verb. Any entry with irregular morphological behaviour will also include the features :PLURAL, :PAST and so on, with the relevant values. All verbs have a :SUBC feature and for our purposes this is the most interesting feature. In the case of the example above, the subcategorisation values specify that *reimburse* can occur with two object noun phrases (NP-NP), an object noun phrase followed by a prepositional phrase headed by "for" (NP-PP :PVAL ("for")) or just an object noun phrase (NP).[3] What makes the COMLEX resource particularly suitable for our evaluation is that each of the complement types (NP-NP, NP-PP and NP) which make up the value of the :SUBC feature is associated with a formal frame definition which looks as follows:

```
(vp-frame  np-np  :cs  ((np 2)(np 3))
                  :gs  (:subject 1 :obj 2 :obj2 3)
                  :ex  "she asked him his name")
```

The value of the :cs feature is the constituent structure of the subcategorisation frame, which lists the syntactic CF-PSG constituents in sequence (omitting the subject,

[3] Note that the details of the subject are not included in COMLEX frames.

again). The value of the :gs feature is the grammatical structure which indicates the functional role played by each of the CF-PSG constituents. The elements of the constituent structure are indexed, and these indices are referenced in the :gs field. The index '1' always refers to the surface subject of the verb. This mapping between constituent structure and functional structure makes the information contained in COMLEX particularly suitable as an evaluation standard for the LFG semantic forms which I induce.

I present the evaluation for the verbs which occur in an active context in the treebank. COMLEX does not provide passive frames. For Penn-II (WSJ), there are 2847 verb lemmas (used actively) that both resources have in common. 2815 verb lemmas appear in COMLEX but not in the induced lexicon and 290 verb lemmas (used actively) appear in the induced lexicon but not in COMLEX (Figure 4.1). For Penn-III (WSJ and Brown), COMLEX and the induced lexicon share 3535 verb lemmas (used actively). This is shown in Figure 4.2.[4]

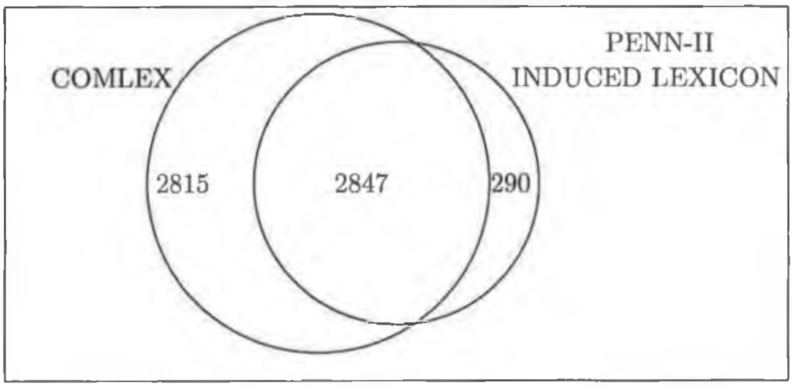


Figure 4.1: Intersection between Active Verb Lemma Types in COMLEX and the Penn-II Induced Lexicon

[4] Given these figures, one may begin to wonder about the value of automatic induction. First, COMLEX does not rank frames by probabilities, which are essential in disambiguation. Second, the coverage of COMLEX is not complete: 527 lemmas "discovered" by the induction experiment are not listed in COMLEX; see error analysis in Section 4.3.5 below. Third, there are frame types not attested in COMLEX but "discovered" by our method.

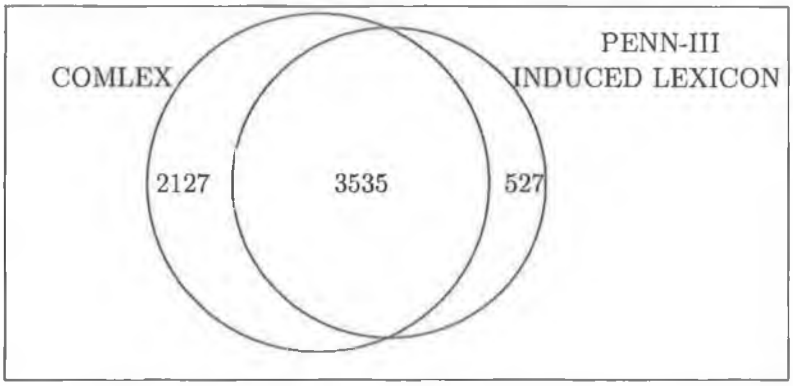


Figure 4.2: Intersection between Active Verb Lemma Types in COMLEX and the Penn-III Induced Lexicon

4.4.2 COMLEX-LFG Mapping I

In order to carry out the evaluation, I have to find a common format for the expression of subcategorisation information between the induced LFG-style subcategorisation frames and those contained in COMLEX. The following are the common syntactic functions: SUBJ, OBJ, OBJ$_i$, COMP and PART. Unlike our system, COMLEX does not distinguish an OBL from an OBJ$_i$ so I converted all the obliques in the induced frames to OBJ$_i$. As in COMLEX, the value of i depends on the number of objects/obliques already present in the semantic form. COMLEX does not differentiate between COMPs and XCOMPs as our system does (control information is expressed in a different way, see Section 4.3.3), so I conflate the two LFG categories to that of COMP. The process is summarised in Table 4.6.

Induced Functions	COMLEX	Merged
SUBJ	Subject	SUBJ
OBJ	Object	OBJ
OBJ2	Obj2	OBJ$_i$
OBL	Obj3	
OBL2	Obj4	
COMP	Comp	COMP
XCOMP		
PART	Part	PART

Table 4.6: Mapping I - Merging of COMLEX and LFG Syntactic Functions

The manually constructed COMLEX entries provide a gold standard against which I evaluate the automatically induced frames. I calculate the number of true positives (*tp*) (where the induced semantic forms and those from COMLEX are the same), the number of false negatives (*fn*) (those frames which appeared in COMLEX but were not produced by our system) and the number of false positives (*fp*) (those frames produced by our system which do not appear in COMLEX). I calculate precision, recall and f-score using the following standard equations:

$$precision = \frac{tp}{tp + fp}$$

$$recall = \frac{tp}{tp + fn}$$

$$\textit{f-score} = \frac{2 * recall * precision}{recall + precision}$$

This scores are then averaged over the verbs being evaluated. I use the frequencies associated with each of the semantic forms in order to set a *relative* threshold to filter the selection of semantic forms. For a threshold of 1% I disregard any semantic forms with a conditional probability (i.e. given a lemma) of less than or equal to 0.01.[5] As some verbs occur less frequently than others, I think it is important to use a relative rather than absolute threshold (as (Carroll and Rooth, 1998), for instance) in this way. I carried out the evaluation in a similar way to (Schulte im Walde, 2002b), the only experiment comparable in scale to ours. The figures shown in Table 4.7 give the results of three different experiments with the relative threshold set to 1%. As for all the results tables, the baseline figures (simply assigning the most frequent frames, in this case transitive and intransitive, to each lemma by default) are in each case shown in the left column while the results achieved by the induced lexicon are presented in the right column. Distinguishing between complement and adjunct prepositional phrases is a notoriously difficult aspect of automatic subcategorisation frame acquisition. For this reason, following the evaluation setup in (Schulte im Walde, 2002b), the three experiments vary with respect to the amount

[5] In this chapter two arbitrarily chosen thresholds (1% and 5%) are used for each evaluation, with the exception of an experiment described in Section 4.4.4 where absolute thresholds of 100 and 200 are used.

Mapping I	Precision		Recall		F-Score	
	Baseline	**Induced**	**Baseline**	**Induced**	**Baseline**	**Induced**
Exp. 1	66.1%	75.2%	65.8%	69.1%	66.0%	72.0%
Exp. 2	71.5%	65.5%	64.3%	63.1%	67.7%	64.3%
Exp. 3	64.7%	71.8%	11.9%	16.8%	20.1%	27.3%

Table 4.7: Results of Penn-II Active Frames (2847) Evaluation against COMLEX (Relative Threshold of 1%)

of prepositional information contained in the subcategorisation frames.

Experiment 1: Here I excluded subcategorised prepositional phrase arguments entirely from the comparison. In a manner similar to that of (Schulte im Walde, 2002b), any frames containing an OBL were mapped to the same frame type minus that argument. For example, the frame `[subj,obl:for]` becomes `[subj]`. Using a relative threshold of 1% (Table 4.7), precision is 75.2%, recall 69.1% and f-score 72.0%.

Experiment 2: Here I include subcategorised prepositional phrase arguments but only in their simplest form, i.e. they are not parameterised for particular prepositions. For example, the frame `[subj,obl:for]` is rewritten as `[subj,obl]`. Using a relative threshold of 1% (Table 4.7), precision is 65.5%, recall 63.1% and f-score 64.3%.

Experiment 3: Here I used semantic forms which contain details of specific prepositions for any subcategorised prepositional phrase (e.g. `[subj,obl:for]`). Using a relative threshold of 1% (Table 4.7), the precision figure (71.8%) is quite high. However the recall (16.8%) is very low. Consequently the f-score (27.3%) is also low. The reason for this is discussed in Section 4.3.4.

Interestingly my results are very similar to those reported by Schulte im Walde (2002b). However, due to the difference in approach (parser-based versus treebank-based), language (German versus English), gold standard (Duden versus COMLEX) and quantity of source data (18.7 million words versus 1 million words), the results are not really comparable.

The figures in Table 4.8 are the result of the second experiment where the relative

Mapping I	Precision		Recall		F-Score	
	Baseline	**Induced**	**Baseline**	**Induced**	**Baseline**	**Induced**
Exp. 1	66.1%	80.2%	65.8%	63.6%	66.0%	70.9%
Exp. 2	71.5%	69.6%	64.3%	56.9%	67.7%	62.7%
Exp. 3	64.7%	76.7%	11.9%	13.9%	20.1%	23.5%

Table 4.8: Results of Penn-II Active Frames (2847) Evaluation against COMLEX (Relative Threshold of 5%)

threshold was increased to 5%. The effect of such an increase is obvious in that precision goes up (by as much as 5%) for each of the three evaluations while recall goes down (by as much as 5.5%). This is to be expected as a higher threshold means that there are fewer semantic forms associated with each verb in the induced lexicon but they are more likely to be correct due to their greater frequency of occurrence. The conditional probabilities I associate with each semantic form together with thresholding can be used to customise the induced lexicon to the task for which it is required, i.e. whether a very precise lexicon is preferred to one with broader coverage. In Tables 4.7 and 4.8, the baseline is exceeded in all experiments with the exception of Experiment 2 (67.7% versus 62.7% f-score). This is attributed to Mapping I where OBL$_i$ becomes OBJ$_i$ (Table 4.6) and the particulars of Experiment 2 which includes obliques without the specific preposition, i.e. the frame `[subj,obj:with]` becomes `[subj,obj]`. Therefore, the transitive baseline frame scores better than it should against the gold standard. A more fine-grained LFG-COMLEX Mapping II where this effect disappears is presented in Section 4.4.3.

Induced Functions	COMLEX	Merged
SUBJ	Subject	SUBJ
OBJ	Object	OBJ
OBJ2	Obj2	OBJ2
OBL	Obj3	OBL
OBL2	Obj4	OBL2
COMP	Comp	COMP
XCOMP	Comp	XCOMP
PART	Part	PART

Table 4.9: Mapping II - Merging of COMLEX and LFG Syntactic Functions

4.4.3 COMLEX-LFG Mapping II and Penn-II

The COMLEX-LFG Mapping I presented above establishes a 'least common denominator' for the COMLEX and the LFG-inspired resources. More fine-grained mappings are possible: in order to ensure that the mapping from our semantic forms to the COMLEX frames did not oversimplify the information in the automatically extracted subcategorisation frames, I conducted a further set of experiments where I converted the information in the COMLEX entries to the format of the extracted semantic forms. I explicitly differentiated between OBLs and OBJs in the COMLEX source data-structures by automatically deducing if a COMLEX OBJ$_i$ was co-indexed with an np or a pp. Furthermore, as can be seen in the following example, COMLEX frame definitions contain details of the control patterns of sentential complements, encoded using the ':features' attribute. This allows for the automatic discrimination between COMPs and XCOMPs.

```
(vp-frame   to-inf-sc   :cs          (vp 2 :mood to-infinitive :subject 1)
                        :features    (:control subject)
                        :gs          (:subject 1 :comp 2)
                        :ex          "I wanted to come")
```

The mapping is summarised in Table 4.9. The results of the subsequent evaluation are presented in Table 4.10 and Table 4.11. I have added Experiments 2a and 3a. These are the same as Experiments 2 and 3 except that they additionally include the specific particle with each PART function. While the recall figures in Tables 4.10 and 4.11 are slightly lower than those in Tables 4.7 and 4.8 with the exception of Experiment 3, changing the mapping in this way results in an increase in precision in each case (by as much as 14.1%). The results of the lexical evaluation are consistently better than the baselines, in some cases by 15% and over (Experiments 2 and 2a, threshold 1%). Notice that in contrast to Tables 4.7 and 4.8, in the more fine-grained COMLEX-LFG Mapping-II presented here, all experiments exceed the baseline.

Mapping II	Precision		Recall		F-Score	
	Baseline	**Induced**	**Baseline**	**Induced**	**Baseline**	**Induced**
Exp. 1	71.9%	83.5%	60.3%	63.3%	65.6%	72.0%
Exp. 2	65.2%	79.6%	37.6%	53.2%	47.7%	63.8%
Exp. 2a	65.2%	77.7%	32.8%	45.9%	43.6%	57.6%
Exp. 3	65.2%	74.2%	15.3%	23.1%	24.7%	35.2%
Exp. 3a	65.2%	73.5%	13.6%	20.7%	22.5%	32.3%

Table 4.10: Results of Penn-II Active Frames (2847) Evaluation against COMLEX (Relative Threshold of 1%)

Mapping II	Precision		Recall		F-Score	
	Baseline	**Induced**	**Baseline**	**Induced**	**Baseline**	**Induced**
Exp. 1	71.9%	87.7%	60.3%	58.4%	65.6%	70.1%
Exp. 2	65.2%	83.7%	37.6%	46.7%	47.7%	60.0%
Exp. 2a	65.2%	82.2%	32.8%	40.0%	43.6%	53.7%
Exp. 3	65.2%	79.0%	15.3%	18.8%	24.7%	30.4%
Exp. 3a	65.2%	78.5%	13.6%	16.6%	22.5%	27.4%

Table 4.11: Results of Penn-II Active Frames (2847) Evaluation against COMLEX (Relative Threshold of 5%)

4.4.4 Penn-III (Mapping-II)

I also extract a lexicon from the Penn-III Treebank, a more balanced corpus resource. Penn-III consists of the WSJ section from Penn-II as well as a parse-annotated subset of the Brown Corpus. The subset of the Brown Corpus comprises 24,242 trees compiled from a variety of text genres. It has been shown (Roland and Jurafsky, 1998) that the subcategorisation tendencies of verbs vary across linguistic domains. The aim, therefore, is to increase the scope of the induced lexicon not only in terms of the verb lemmas for which entries exist, but also in terms of the frames with which they co-occur. As described in the previous chapter, the f-structure annotation algorithm was extended with only minor amendments to cover the parsed Brown Corpus.

When evaluating the application of the lexical extraction system on Penn-III I carried out two sets of experiments, identical in each case to those described for Penn-II, including the use of relative (1% and 5%) rather than absolute thresholds. For the first set of experiments I evaluated the lexicon induced from the parse-annotated Brown Corpus only. This evaluation was for 2719 active verb lemmas using the more fine-grained Mapping-II.

Mapping II	Precision		Recall		F-Score	
	Baseline	**Induced**	**Baseline**	**Induced**	**Baseline**	**Induced**
Exp. 1	73.2%	80.9%	60.2%	61.4%	66.0%	69.8%
Exp. 2	66.0%	71.7%	37.7%	51.9%	48.0%	60.2%
Exp. 2a	66.0%	70.0%	32.7%	44.5%	43.8%	54.4%
Exp. 3	66.0%	62.0%	14.7%	22.2%	24.1%	32.7%
Exp. 3a	66.0%	61.9%	13.1%	20.0%	21.9%	30.2%

Table 4.12: Results of Penn-III Active Frames (2719) (Brown Corpus Only) COMLEX Comparison (Relative Threshold of 1%)

Mapping II	Precision		Recall		F-Score	
	Baseline	**Induced**	**Baseline**	**Induced**	**Baseline**	**Induced**
Exp. 1	73.2%	84.6%	60.2%	57.5%	66.0%	68.5%
Exp. 2	66.0%	76.7%	37.7%	47.3%	48.0%	58.5%
Exp. 2a	66.0%	75.6%	32.7%	40.4%	43.8%	52.7%
Exp. 3	66.0%	66.8%	14.7%	17.9%	24.1%	28.2%
Exp. 3a	66.0%	66.6%	13.1%	15.9%	21.9%	25.7%

Table 4.13: Results of Penn-III Active Frames (2719) (Brown Corpus Only) COMLEX Comparison (Relative Threshold of 5%)

Tables 4.12 and 4.13 show that the results generally exceed the baseline, in the case of Experiment 2 by over 12%, similar to those recorded for Penn-II (Tables 4.10 and 4.11). While the precision is slightly lower than that reported for the experiments in Tables 4.10 and 4.11, in particular for Experiments 2 to 3a where details of obliques are included, the recall in each of these experiments is slightly higher than that recorded for Penn-II. I conjecture that the main reasons for this are twofold: first, the amended approach to the annotation of obliques is slightly less precise and conservative than the largely -CLR tag-driven approach taken for Penn-II, and second, the parse-annotated section of the Brown Corpus exhibits considerably more domain variation than the newspaper material in the WSJ sections. Consequently I record an increase in recall and a drop in precision. This trend is repeated in the second set of experiments. In this instance, I combined the lexicon extracted from the WSJ with that extracted from the parse-annotated Brown Corpus and evaluated the resulting resource, for 3535 active verb lemmas. The results are shown in Tables 4.14 and 4.15. The results compare very positively against the baseline. The Precision scores are lower than those reported for Penn-II (Tables 4.10 and 4.11) but higher than those for the Brown Corpus evalation (Tables 4.12 and 4.13). There is a consistent increase in recall and an overall increase in f-score (by up to 3.3%). Using Noreen's (1989) Approximate Randomization Test, I test for statistical significance of the results established for the largest treebank-based extraction of lexical resources against the baseline (cf. Table 4.14). In the case of each experiment, the induced lexicon outperforms the baseline significantly at the 95% confidence level.

Mapping II	**Precision**		**Recall**		**F-Score**	
	Baseline	**Induced**	**Baseline**	**Induced**	**Baseline**	**Induced**
Exp. 1	71.3%	78.8%	62.7%	68.0%	66.7%	73.0%
Exp. 2	64.6%	72.0%	39.6%	59.5%	49.1%	65.2%
Exp. 2a	64.6%	70.0%	35.0%	51.9%	45.4%	59.6%
Exp. 3	64.6%	64.3%	16.2%	26.6%	26.0%	37.7%
Exp. 3a	64.6%	64.0%	14.6%	24.1%	23.8%	35.1%

Table 4.14: Results of Penn-III Active Frames (3535) (Brown and WSJ) COMLEX Comparison (Relative Threshold of 1%)

Mapping II	**Precision**		**Recall**		**F-Score**	
	Baseline	**Induced**	**Baseline**	**Induced**	**Baseline**	**Induced**
Exp. 1	71.3%	83.9%	62.7%	62.2%	66.7%	71.4%
Exp. 2	64.6%	77.8%	39.6%	52.2%	49.1%	62.5%
Exp. 2a	64.6%	76.6%	35.0%	45.3%	45.4%	57.0%
Exp. 3	64.6%	70.2%	16.2%	20.7%	26.0%	32.0%
Exp. 3a	64.6%	69.8%	14.6%	18.5%	23.8%	29.2%

Table 4.15: Results of Penn-III Active Frames (3535) (Brown and WSJ) COMLEX Comparison (Relative Threshold of 5%)

about	across	along	around
behind	below	beneath	between
beyond	by	down	from
in	inside	into	off
on	onto	out	out of
outside	over	past	through
throughout	to	toward	towards
up	up to	via	

Table 4.16: COMLEX Directional Prepositions

Directional Prepositions

A consistent pattern that emerges when viewing the results presented here is that of a sharp drop in recall for Experiments 3 and 3a with a consequent drop in f-score. For example, the recall figures for Experiments 3 and 3a in Table 4.14 (26.6% and 24.1%) are considerably lower than those for Experiments 2 and 2a (59.5% and 51.9%). This can be accounted for by the fact that the creators of COMLEX have chosen to err on the side of overgeneration as regards the list of prepositions which may occur with a verb and a subcategorisation frame containing a prepositional phrase. This is particularly true of directional prepositions. For COMLEX, a list of 31 directional prepositions (Table 4.16) was prepared and assigned in its entirety by default to any verb which can potentially appear with any directional preposition in order to save time and avoid the risk of missing prepositions. Grishman et al. (1994) admit that this can lead to a preposition list which is a 'little rich' for a particular verb but this is the approach they have chosen to take. In a subsequent experiment, I incorporated this list of directional prepositions by default into the semantic form induction process in the same way as the creators of COMLEX have done: if a verb is a COMLEX p-dir verb *and* it has a frame containing an OBL with one of the prepositions in the p-dir list, then it is assigned the entire list of 31 prepositions. The results are presented in Table 4.17. The effect was as expected: the recall score for the two experiments increased to 50.3% and 45.3% (from 26.6% and 24.1%) while the f-scores increased to 58.3% and 54.7% (from 37.7% and 35.1%). This experiment does not tell us anything new about the quality of the induced lexicon *per se* but attempts to account for the low recall scores for Experiments 3 and 3a.

Mapping II	**Precision**	**Recall**	**F-Score**
Exp. 3	69.4%	50.3%	58.3%
Exp. 3a	68.9%	45.3%	54.7%

Table 4.17: Penn-III Active Frame Evaluation (3535) against COMLEX using P-Dir List (Relative Threshold of 1%)

Passive Evaluation

Table 4.18 presents the results of the evaluation of the passive semantic forms extracted. It was carried out for 2430 verb lemmas in Penn-III which occur with passive frames and also occur in COMLEX. COMLEX does not provide passive frames. I applied Lexical Redundancy Rules (Kaplan and Bresnan, 1982) to automatically convert the active COMLEX frames to their passive counterparts: e.g. subjects are demoted to optional 'by' oblique agents while direct objects become subjects. The resulting precision and recall scores were high, achieving f-scores of 68.5% and 62.5% for Experiments 2 and 2a.

In a second experiment, I extended the induced lexicon by applying reversed Lexical Redundancy Rules to the automatically extracted passive semantic forms. This extended lexicon was again evaluated against COMLEX. The evaluation was for 3673 verb lemmas (an increase of 138 lemmas on the previous Penn-III evaluation). The results are shown in Table 4.19. Although the evaluation is for a larger number of verbs there has not been a detrimental effect on lexicon quality. The f-scores for Experiments 2 and 2a (65.4% and 59.9%) are a slight improvement on the original Penn-III evaluation (Table 4.14). With a slight drop in precision and a slight increase in recall, the f-scores for Experiments 3 and 3a are unchanged.

Passive	**Precision**	**Recall**	**F-Score**
Experiment 2	75.5%	62.7%	68.5%
Experiment 2a	74.3%	53.9%	62.5%

Table 4.18: Results of Penn-III Passive Frame Evaluation (Relative Threshold of 1%)

Passive	**Precision**	**Recall**	**F-Score**
Experiment 2	71.7%	60.1%	65.4%
Experiment 2a	69.8%	52.5%	59.9%
Experiment 3	63.5%	26.8%	37.7%
Experiment 3a	63.2%	24.3%	35.1%

Table 4.19: Results of Evaluation of Penn-III Lexicon Extended with Lexical Redundancy Rules (Relative Threshold of 1%)

Absolute Thresholds

Many of the previous approaches discussed in Chapter 2 use a limited number of verbs for evaluation, based on their absolute frequency in the corpus. I introduced absolute thresholds to the filtering phase of my lexical extraction system to examine the effect on quality. Table 4.20 shows the results of Experiment 2 for all verbs, for the verb lemmas with an absolute frequency greater than 100 and for verbs with a frequency greater than 200. Unsurprisingly, the use of an absolute threshold leads to an increase in precision (from 72.0% to 79.0% and 79.3%), an increase in recall (from 59.5% to 65.9% to 64.4%), and an overall increase in f-score (from 65.2% to 71.9% and 71.1%). Increasing the threshold from 100 to 200 resulted in a slight drop in f-score. High frequency verbs tend to be associated with a larger number of subcategorisation frames (including very rare frames) than lower frequency verbs. It is therefore unsurprising that our system performs worse in a qualitative evaluation focusing on these more challenging verbs.

Threshold	**Precision**	**Recall**	**F-Score**
All	72.0%	59.5%	65.2%
Threshold 100 (274)	79.0%	65.9%	71.9%
Threshold 200 (142)	79.3%	64.4%	71.1%

Table 4.20: Penn-III Active Frames Evaluation against COMLEX using Absolute Thresholds (Exp. 2)

4.4.5 Error Analysis and Discussion

The work presented in this section highlights a number of issues associated with the evaluation of automatically induced subcategorisation frames against an existing external gold standard, in this case COMLEX. While this evaluation approach is arguably less labour-intensive than the manual construction of a custom-made gold standard, it does introduce a number of difficulties into the evaluation procedure. It is a non-trivial task to convert both the gold standard and the induced resource to a common format in order to facilitate fair and comprehensive evaluation. In addition, as the results show, the choice of common format and mapping to it can affect the results. In COMLEX-LFG Mapping I (Section 4.3.2), I found that mapping from the induced lexicon to COMLEX resulted in higher

Recall scores than those achieved when I (effectively) reversed the mapping (COMLEX-LFG Mapping II (Section 4.3.3)). The first mapping is essentially a conflation of the more fine-grained LFG grammatical functions to the more generic COMLEX functions, while the second mapping tries to maintain as many distinctions as possible.

Another substantial drawback to using an existing external gold standard such as COMLEX to evaluate an automatically induced subcategorisation lexicon is that the resources are not necessarily constructed from the same source data. I have extracted frames from two sources (the WSJ and the Brown Corpus) whereas COMLEX was built using examples from the San Jose Mercury News, the Brown Corpus, several literary works from the Library of America, scientific abstracts from the U.S. Department of Energy, and the WSJ. For this reason, it is likely to contain a greater variety of subcategorisation frames than the induced lexicon. It is also possible that due to human error COMLEX contains subcategorisation frames, the validity of which are in doubt, for example the overgeneration of subcategorised-for directional prepositional phrases. This is due to the fact that the aim of the COMLEX project was to construct as complete a set of subcategorisation frames as possible, even for infrequent verbs. Lexicographers were allowed to extrapolate from the citations found, a procedure which is bound to be less certain than the assignment of frames based entirely on existing examples. (Briscoe, 2001) notes that lexicons such as COMLEX tend to demonstrate high precision but low recall. Briscoe and Carroll (1997) report on manually analysing an open-class vocabulary of 35,000 head words for predicate subcategorisation information and comparing the results against the subcategorisation details in COMLEX. Precision was quite high (95%) but recall was low (84%). This has an effect on both the precision and recall scores of our system against COMLEX. In order to ascertain the effect of using COMLEX as a gold standard for the induced lexicon, I carried out some more detailed error analysis, the results of which are summarised in Table 4.21. I randomly selected 80 false negatives (fn) and 80 false positives (fp) across a range of active frame types containing prepositional and particle detail taken from Penn-III and manually examined them in order to classify them into 'correct' and 'incorrect'. Of the 80 fps, 33 were manually judged to be legitimate subcategorisation frames. For example, as Table 4.21 shows, there are a number of correct transitive verbs ([subj,obj]) in the

Frame Type	COMLEX: fn		Induced: fp	
	Correct	Incorrect	Correct	Incorrect
[subj]	9	1	4	6
[subj,obj]	10	0	9	1
[subj,obj,obj2]	7	3	1	9
[..,xcomp,..]	10	0	1	10
[..,comp,..]	7	3	4	5
[..,obl,..]	23	7	14	16

Table 4.21: COMLEX Error Analysis

automatically induced lexicon which are not included in COMLEX. This examination was also useful in highlighting the frame types on which the lexical extraction procedure was performing poorly, in our case those containing XCOMPs and those containing OBJ2s. Out of 80 fns, 14 were judged to be incorrect when manually examined. These can be broken down as follows: one intransitive frame, three ditransitve frames, three frames containing a COMP and seven frames containing an OBL were found to be invalid.

4.5 OALD Evaluation

I also evaluate the induced lexical resources against a machine-readable version of the Oxford Advanced Learner's Dictionary (Hornby, 1980). I describe this resource and contrast its content and encoding with that of COMLEX. I then outline the mapping between the OALD gold standard and the induced lexicon including the extensions which had to be made to the extraction procedure to incorporate the required information into the extracted frames. Finally I present and discuss the evaluation results.

4.5.1 OALD

The machine-readable version of the OALD was originally produced in 1986 by Richard Mitton. I use an extended version released in 1992 for evaluation.[6] It contains 70646 entries including inflected forms. An entry comprises a graphemic representation, a phonemic representation, one or more part-of-speech tags with rarity flags, a syllable count and a set of verb patterns for verbs. The following is a sample entry for the uninflected form of

[6] This resource is available from http://ota.ahds.ac.uk/.

the verb *zip*:

zip zIp H4%,K6% 16A,15B,22

The first field in the entry contains the spelling of the word in question and the second field its pronunciation. The third field contains the syntactic tags of the entry. The first character of the syntactic tag indicates that *zip* can be a transitive verb (H) or a countable noun (K). The second character supplies inflectional information. The inflection codes 0 to 5 are for verbs. 4 denotes the following inflectional behaviour: "stem+s; double final letter +ing or +ed". 6 is an nominal inflection code and states that 's' is added to form the plural. The % sign on both syntactic tags is a so-called 'rarity' flag and indicates that the word (used as that part-of-speech) is neither extremely common (it does not occur in a list of the most frequent 500 words in a word list compiled from a number of corpora) nor extremely rare. The final field contains the number of syllables in the word (1) and, in the case of verbs, the 'verb patterns' or sentential constructions in which that verb can occur. 6A is a simple transitive verb which may be passivised, for example *He zipped his jacket.* The pattern 15B comprises a direct object followed by an adverbial particle or an adverbial particle followed by the direct object, for example *He zipped up his coat.* Pattern 22 is where the direct object is followed by an adjective which indicates result or manner, for example *He zipped his coat shut.* For the purposes of evaluation, I am only interested in verbal entries. In total there are entries for 5771 uninflected verbs. I evaluate the overlap of 3466 verb lemmas (extracted from Penn-III) between the induced lexicon and the OALD (Figure 4.3).

4.5.2 Mapping

There are 51 verb patterns used in the OALD verbal entries. These are referenced in the dictionary using an index associated with a description containing examples. In contrast to COMLEX, the patterns associated with a verb in an OALD entry may contain adjuncts as well as arguments. Verb patterns are, therefore, not equivalent to subcategorisation frames. As part of the mapping process, it was necessary to distinguish patterns encoding information about verbal arguments from those containing adjunct information. The complete list of verb patterns is presented in Table 4.22. For evaluation purposes, it was

Code	Verb Pattern	Example	Mapped Frame
VP1	S+BE+subj complement/adjunct There/It+BE+S	*The children are asleep* *There was a large crowd*	[subj,xcomp(nv)]
VP2A	S+vi	*The moon rose*	[subj]
VP2B	S+vi (for)+ adverbial adjunct	*We walked (for) five miles*	[subj]
VP2C	S+vi+adverbial adjunct/particle	*I'll catch up with you*	[subj,part:adval]
VP2D	S+vi+adjective/noun/pronoun	*He died a millionaire*	[subj]
VP2E	S+vi+pres. participle	*She lay smiling at me*	[subj]
VP3A	S+vi+prep+noun/pronoun	*You can rely on that man*	[subj,obl:pval]
VP3B	S+vi+(prep(+*it*))+clause	*I agree that it was a mistake*	[subj,comp]
VP4A	S+vi+*to*-inf	*We stopped to rest*	[subj,xcomp(toinf)]
VP4B	S+vi+*to*-inf	*He awoke to find the house on fire*	[subj]
VP4C	S+vi+*to*-inf	*He agreed to come*	[subj,xcomp(toinf)]
VP4D	S+SEEM/APPEAR+(*to be*)adjective/noun *It*+SEEM/APPEAR+adjective/noun+S	*This seems (to be) a serious matter* *It seems a pity to waste that food*	[subj,xcomp(tobe)]
VP4E	S+SEEM/APPEAR/HAPPEN/CHANCE +*to*-inf	*The baby seems to be asleep*	[subj,xcomp(toinf)]
VP4F	S+BE+*to*-inf	*We're to be married in May*	[subj,xcomp(toinf)]
VP5	S+anomalous finite+inf	*You may leave now*	[subj,xcomp(inf)]
VP6A	S+vt+noun/pronoun	*Everyone likes her*	[subj,obj]
VP6B	S+vt+noun/pronoun	*She has green eyes*	[subj,obj]
VP6C	S+vt+gerund	*She enjoys playing tennis*	[subj,xcomp(ing)]
VP6D	S+vt+gerund	*She loves going to the cinema*	[subj,xcomp(ing)]
VP6E	S+NEED/WANT/BEAR+gerund	*My shoes want mending*	[subj,xcomp(ing)]
VP7A	S+vt+(*not*)+*to*-inf	*I forgot to post your letters*	[subj,xcomp(toinf)]
VP7B	S+HAVE/OUGHT+(*not*)+*to*-inf	*You don't have to leave yet*	[subj,xcomp(toinf)]
VP8	S+vt+interrogative pronoun/adverb+*to*-inf	*Do you know how to do it?*	[subj,obj(intinf)]
VP9	S+vt+*that*-clause	*I think that it'll rain*	[subj,comp]
VP10	S+vt+dependent clause/question	*She asked why I was late*	[subj,obj(int)]
VP11	S+vt+noun/pronoun+*that*-clause	*He warned us that the roads were icy*	[subj,obj,comp]
VP12A	S+vt+IO+DO	*He doesn't owe me anything*	[subj,obj,obj2]
VP12B	S+vt+IO+DO	*She cooked her husband some sausages*	[subj,obj,obj2]
VP12C	S+vt+noun/pronoun+noun/pronoun	*Ask him his name*	[obj,obj2]
VP13A	S+vt+DO+*to*+noun/pronoun	*I've sent presents to my family*	[subj,obj,obl:to]
VP13B	S+vt+DO+*for*+noun/pronoun	*Will you do a favour for me?*	[subj,obj,obl:for]
VP14	S+vt+DO+prep+noun	*I explained my difficulty to him*	[subj,obj,obl:pval]
VP15A	S+vt+DO+adverbial phrase	*I put the book on the shelf*	[subj,obj,obl:pval]
VP15B	S+vt+DO+adverbial particle	*Take your shoes off*	[subj,obj,part:adval]
	S+vt+adverbial particle+DO	*She gave away her clothes*	
VP16A	S+vt+DO+*to*-inf	*They left me to do all the work*	[subj,obj]
VP16B	S+vt+DO+*as/like*+noun Subj+vt+DO+*as if/though*+clause	*Her parents spoilt her as a child* *Don't treat her as if she were a servant*	[subj,obj,obl:as]
VP17	S+vt+noun/pronoun+(*not*)+*to*-inf	*They warned us not to be late*	[subj,obj,xcomp(toinf)]
VP18A	S+vt+noun/pronoun+inf	*We felt the house shake*	[subj,obj,xcomp(inf)]
VP18B	S+vt+noun/pronoun+inf	*Let me go*	[subj,obj,xcomp(inf)]
VP18C	S+HAVE+noun/pronoun+inf	*What would you have me do?*	[subj,obj,xcomp(inf)]
VP19A	S+vt+noun/pronoun+pres. participle	*Did you hear me knocking?*	[subj,obj,xcomp(ing)]
VP19B	S+vt+noun/pronoun+pres. participle	*This set me thinking*	[subj,obj,xcomp(ing)]
VP19C	S+vt+noun/pronoun/possessive+pres. participle	*I can't understand him/his being so stupid*	[subj,obj,xcomp(ing)]
VP20	S+vt+noun/pronoun+interrogative+*to*-inf	*Tell him where you put it*	[subj,obj,obj(intinf)]
VP21	S+vt+noun/pronoun+interrogative clause	*Ask him where he put it*	[subj,obj,obj(int)]
VP22	S+vt+DO+adjective	*The sun keeps us warm*	[subj,obj,xcomp(nv)]
VP23	S+vt+DO+noun	*They named the baby Richard*	[subj,obj,xcomp(nv)]
VP24A	S+vt+DO+past participle	*You must make your views known*	[subj,obj,xcomp(nv)]
VP24B	S+HAVE+DO+past participle	*King Charles had his head cut off*	[subj,obj,xcomp(ppart)]
VP24C	S+HAVE/GET+DO+past participle	*Please have/get the programme changed*	[subj,obj,xcomp(ppart)]
VP25	S+vt+DO+(*to be*)+adjective/noun	*People considered him (to be) innocent*	[subj,obj,xcomp(nv)]

Table 4.22: OALD Frames and their Mappings

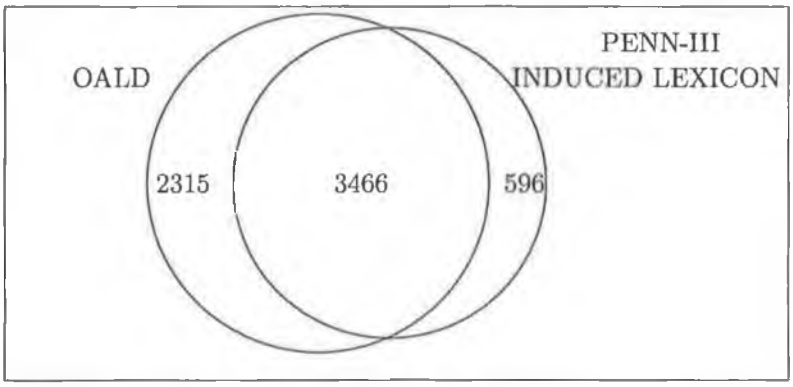


Figure 4.3: Intersection between Active Verb Lemma Types in the OALD and the Penn-III Induced Lexicon

also necessary to find a common representation for the OALD patterns and the induced frames. This is shown in the fourth column of the table. Essentially, the new frames are the original induced frames enriched with extra information to better reflect the information represented in the OALD patterns. There are some subtle differences between OALD verb patterns which could not be translated to the common representation. For example, VP6C and VP6D both have the pattern S+vt+gerund (subject followed by transitive verb followed by gerund). VP6C refers to verbs where the gerund may not be replaced by a to-infinitive whereas for VP6D this substitution may occur. In this case both patterns are mapped to the same frame. Changes were made to the semantic form extraction procedure to incorporate extra information where possible into the extracted lexicon:

- XCOMPs: To map between the OALD verb patterns and the induced frames, I enrich the basic `xcomp` function at extraction time to match the more fine-grained XCOMP distinctions made by the OALD. If the `xcomp` contains the `to_inf:+` feature value pair, then the function is rewritten as `xcomp(toinf)`. If in addition its `pred` value is `be`, it is rewritten as `xcomp(tobe)`. Likewise if the `xcomp` contains the `participle:pres` feature value pair, it is rewritten as `xcomp(ing)`. If the `xcomp` contains the `ppart:+` attribute and value, it is rewritten as `xcomp(ppart)`. If the `xcomp` does not contain a `to_inf` feature, a `modal` feature, a `tense` feature nor a `participle` feature but does contain a `cat:v` attribute value pair, it is rewritten as `xcomp(inf)`. Finally, if the `xcomp` does not contain a `cat:v` feature value pair, it is rewritten as `xcomp(nv)`.

- COMPs and OBJs: If the f-structure value of a grammatical function (generally COMP or OBJ) contains a `focus` feature and its interrogative pronoun is a member of a predefined set of question words, then it is rewritten to `obj(int)`. If in addition the f-structure contains the `to_inf:+` feature value pair, then the grammatical function is rewritten as `obj(intinf)`.

Diathesis Alternations

In general, the OALD frames do not contain information about the specific preposition used in a subcategorised for PP. Therefore I conflate the specific prepositions in the extracted frames to the general `pval` variable. There are two exceptions in the OALD, however: `[obj,obl:to]` (13A) and `[obj,obl:for]` (13B). These frames refer to a special case where a ditransitive verb such as *give* can express its second object in an alternative way. The alternation from `give[obj,obj2]` to `give[obj,obl:to]` is an example of the dative alternation. The alternation from `get[obj,obj2]` to `get[obj,obl:for]` is an example of the benefactive alternation. The system originally treated all prepositions in the same way but because of the two gold standard frames 13A and 13B, I had to find a way of dealing with these special cases.

Simply using frequency thresholds was not an option as it would potentially lead to incorrect mappings as in the following example: *He flew the plane to France.* Extracting the frame `fly[obj,obl:to]` in this case would be incorrect and would result not only in a false positive but also a false negative as it should be mapped to the frame `[obj,obl:pval]`.

There has been some work on the automatic acquisition of verb alternation information and its use in subcategorisation frame extraction (Korhonen, 1998; McCarthy and Korhonen, 1998; Lapata, 1999). For the purpose of evaluation, I developed the following simple algorithm based on linguistic knowledge which is applied prior to filtering to overcome the issue associated with these frames:
For each verb, *v*:

```
if $\mathcal{P}(f_2|v) > 0$ && $\mathcal{P}(f_1|v) < \theta$
then
    $\mathcal{P}(f_3|v) = \mathcal{P}(f_3|v) + \mathcal{P}(f_2|v)$
    $\mathcal{P}(f_2|v) = 0$
```

where f_1 is `[obj,obj2]`, f_2 is the alternative frame (`[obj,obl:for]` or `[obj,obl:to]`), f_3 is `[obj,obl:pval]` and θ is a reliability threshold for the ditransitive frame. I use a threshold of 0.01.

4.5.3 Results

I carried out three experiments for 3466 active verb lemmas extracted from Penn-III. The results are shown in Table 4.23. In Experiment 1, specific prepositions are not included in any of the frames. The generic **pval** is used instead. For Experiment 2, the specific prepositions are included for frames 13A and 13B. The frames are filtered using a relative frequency threshold as before. As expected, including the prepositional detail results in a drop in f-score (by 2%). To improve this score, I use the alternation algorithm as described above (Experiment 3). The f-score is 1.6% higher than the f-score for Experiment 2 and just 0.4% lower than Experiment 1 which does not include any prepositional information.

	Precision		**Recall**		**F-Score**	
	Baseline	**Induced**	**Baseline**	**Induced**	**Baseline**	**Induced**
Exp. 1	64.4%	67.5%	49.5%	61.8%	56.0%	64.5%
Exp. 2	64.4%	64.5%	49.5%	60.6%	56.0%	62.5%
Exp. 3	64.4%	67.2%	49.5%	61.2%	56.0%	64.1%

Table 4.23: Evaluation of Penn-III Lexicon (3466) against OALD (Relative Threshold of 1%)

Error Analysis and Discussion

As for COMLEX, I carried out a more detailed error analysis of the OALD evaluation. Again I randomly selected 80 fns and 80 fps across a range of active frame types and manually classified them as 'correct' or 'incorrect'. The results are summarised in Table 4.24. Of the 80 fps, 32 were manually judged to be legitimate verb frame combinations. As for COMLEX, a large portion (7 out of 10) of correct transitive frames in the induced lexicon

Frame Type	OALD: fn		Induced: fp	
	Correct	Incorrect	Correct	Incorrect
[subj]	6	4	1	9
[subj,obj]	9	1	7	3
[subj,obj,obj2]	7	3	4	6
[..,xcomp,..]	7	3	4	6
[..,comp,..]	3	7	4	6
[..,obl,..]	19	11	12	18

Table 4.24: OALD Error Analysis

were not found in the OALD gold standard. These include the following verbs: appeal (e.g. Commonwealth Edison said it is already appealing the underlying commission order and is considering appealing Judge Curry's order.), process (e.g. The computer processes 55 million instructions per second and uses only one central processing chip, unlike many rival machines using several processors.), research (e.g. Fear of alienating that judge is pervasive, says Maurice Geiger, founder and director of the Rural Justice Center in Montpellier, Vt., a public interest group that researches rural justice issues.), and tally (e.g. With most legislatures adjourned for the year, small business is tallying its scorecard.). Word sense as well as corpus genre effects the subcategorisation behaviour of verbs. Roland et al. (2000) show that even in a cross-corpus comparison of verbs used with the same sense, there is a visible transitivity shift between WSJ data and mixed genre corpora (Brown and BNC). In general, the WSJ uses verbs in a very business/financial-specific way. This factor goes some way to explaining my findings with regard to valid transitive fps. Out of 80 fns, 33 were found to be incorrect on manual examination. The OALD does not make a clear argument/adjunct distinction as a subcategorisation lexicon such as COMLEX does. For example, of the 10 gold standard comp frames which I examined, 7 were found to be incorrect, including phone([subj,obj,comp]) and drop([subj,comp]).

4.6 Task-Based Evaluation

The final evaluation of the induced lexical resources is an application-based one. In this section, I summarise the work reported in (Cahill et al., 2004b) where the extracted semantic forms are utilised at parse time for LDD resolution. We then carry out an experiment which quantifies the effect of the lexical contribution on parser accuracy.

4.6.1 Parsing with Probabilistic LFG Approximations

In the Chapter 3, I described the automatic annotation algorithm used to assign f-structure information to the Penn Treebank trees prior to lexical extraction. Probabilistic LFG approximations used to parse new text into f-structures are also induced from the treebank resources (Cahill et al., 2002b). Two parsing architectures are used:

- In the **pipeline** architecture, a PCFG extracted from the unannotated treebank data and history-based parsers (Collins, 1999; Charniak, 2000; Bikel, 2002) are used to parse unseen text. The resulting parse trees are annotated by the automatic annotation algorithm and resolved into f-structures.

- In the **integrated** architecture, the treebank is first automatically annotated with f-structure information. Annotated PCFGs (A-PCFG) are then extracted where each non-terminal in the grammar carries an f-structure equation: NP[$\uparrow$OBJ=$\downarrow$] → DT[$\uparrow$SPEC=$\downarrow$] NN[$\uparrow$=$\downarrow$] . A node combined with its annotation is treated as a monadic category. Parsing raw text with an annotated PCFG produces a set of annotated trees. Post-parsing, f-structure equations are collected and resolved into f-structures.

Both architectures parse text into "proto" f-structures, i.e. f-structures with unresolved LDDs, as shown in Figure 4.4.

4.6.2 Long Distance Dependencies in LFG

In LFG, LDDs are resolved at f-structure level using functional uncertainty equations (Dalrymple, 2001), precluding the need for empty nodes and traces at c-structure level.

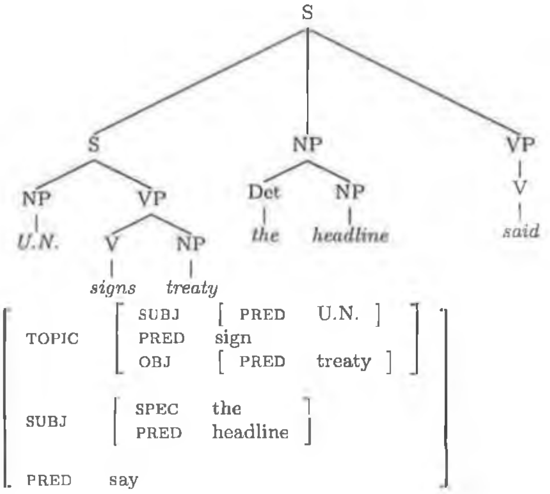


Figure 4.4: Parser Output for String *U.N. signs treaty the headline said* with Unresolved LDD

Functional uncertainty equations are regular expression-based path specifications connecting a source (where linguistic material is encountered) and a target (where the linguistic material is interpreted semantically) position in the f-structure. A functional uncertainty equation of the form ↑TOPIC=↑COMP*COMP is required to account for the fronted sentential constituent in Figure 4.4. The equation states that the value of the TOPIC feature is token identical with the value of the COMP argument which terminates a path through the immediately enclosing f-structure along zero or more COMP attributes. The annotation of the topicalised constituent in the relevant CFG rules is augmented with the functional uncertainty equation:

$$\begin{array}{llll} S \rightarrow & S & NP & VP \\ & \uparrow \text{TOPIC}{=}\downarrow & \uparrow\text{SUBJ}{=}\downarrow & \uparrow{=}\downarrow \\ & \uparrow\text{TOPIC}{=}\uparrow\text{COMP}^{*}\ \text{COMP} & & \end{array}$$

This generates the LDD-resolved proper f-structure in Figure 4.5 for the traceless tree in Figure 5.2.

Aside from functional uncertainty equations, subcategorisation information is a requirement for LFG's account of LDDs. In order for a topicalised element to be resolved as an argument of a local predicate as specified by the functional uncertainty equation, the local predicate must (i) subcategorise for the argument in question and (ii) the argument in

$$\left[\begin{array}{ll}
\textsc{topic} & \left[\begin{array}{ll} \textsc{subj} & \left[\begin{array}{ll}\textsc{pred} & \text{U.N.}\end{array}\right] \\ \textsc{pred} & \text{sign} \\ \textsc{obj} & \left[\begin{array}{ll}\textsc{pred} & \text{treaty}\end{array}\right] \end{array}\right] \\
\textsc{subj} & \left[\begin{array}{ll} \textsc{spec} & \text{the} \\ \textsc{pred} & \text{headline} \end{array}\right] \\
\textsc{pred} & \text{say} \\
\textsc{comp} & \boxed{1}
\end{array}\right]$$

Figure 4.5: F-Structure for String *U.N. signs treaty the headline said* with Resolved LDD

question must not already be filled. As described in Chapter 3, subcategorisation requirements in LFG are enforced by semantic forms and coherence and completeness conditions (all and only the subcategorised-for grammatical functions as expressed by the semantic form must be present). The following are the lexical entries for the verbs in Figures 4.4 and 4.5:

$$\begin{array}{lll} V \rightarrow & \text{said} & \uparrow\textsc{pred}=\text{say}\langle\uparrow\textsc{subj},\uparrow\textsc{comp}\rangle \\ V \rightarrow & \text{signs} & \uparrow\textsc{pred}=\text{sign}\langle\uparrow\textsc{subj},\uparrow\textsc{obj}\rangle \end{array}$$

The subcategorisation requirements of the local pred must be considered in the application of a functional uncertainty equation to ensure that the LDD is resolved at a suitable f-structure location.

4.6.3 Modelling LFG LDD Resolution

Cahill et al. (2004b) describe in detail the design and implementation of an LDD resolution component to create proper f-structures from their 'proto' equivalents. In brief, to model the LFG approach to LDD resolution, two resources are required: semantic forms and encodings of LDD resolution paths (functional uncertainty equations). I describe the automatic acquisition of the lexical resource in Chapter 3. Finite approximations of functional uncertainty equations are acquired by extracting paths between co-indexed material occurring in the automatically generated f-structures from Sections 02-21 of the Penn-II Treebank. 26 unique TOPIC, 60 TOPIC-REL and 13 FOCUS path types are extracted. Each path is assigned a conditional probability. Given a path p and an LDD type t (either TOPIC, TOPIC-REL or FOCUS), the probability of p given t is estimated as:

$$P(p|t) = \frac{count(t,p)}{\sum_{i=1}^{n} count(t,p_i)}$$

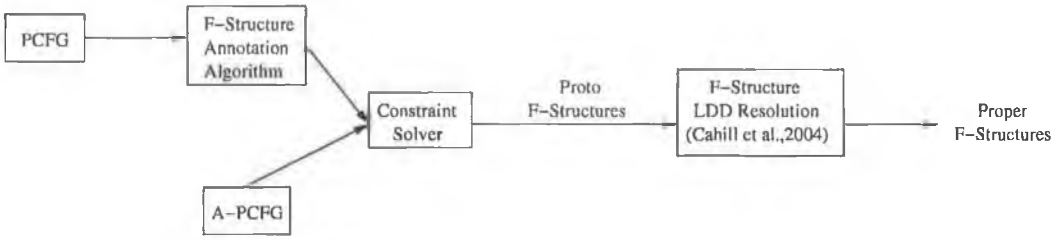

PCFG
F-Structure Annotation Algorithm
Constraint Solver
A-PCFG
Proto F-Structures
F-Structure LDD Resolution (Cahill et al.,2004)
Proper F-Structures

Figure 4.6: Overall Parsing Architecture (Pipeline and Integrated Models) including LDD Resolution Algorithm

Following theoretical LFG, the LDD resolution is applied post-parsing at the f-structure level exploiting the semantic form and path resources as follows. Given a set of semantic forms s with probabilities $P(s|l)$ (where l is a lemma), a set of paths p with probabilities $P(p|t)$ (where t is either TOPIC, TOPIC-REL or FOCUS) and a "proto" f-structure f, the core of the LDD resolution algorithm recursively traverses f to:

```
find TOPIC|TOPIC-REL|FOCUS:$g$ pair
retrieve TOPIC|TOPIC-REL|FOCUS paths
for each path $p$ with $GF_1$:...:$GF_n$:GF traverse
$f$ along $GF_1$:...:$GF_n$ to sub f-structure $h$
retrieve local PRED:$l$
add GF:$g$ to $h$ iff
    • GF is not present at $h$
    • $h$ together with GF is locally complete
      and coherent with respect to a semantic
      form $s$ for $l$
rank resolution by $P(s|l) \times P(p|t)$
```

A distinction is made between active and passive constructions so that the relevant frame type is used at resolution time. The overall parsing architecture is summarised in Figure 4.6.

	Pipeline	Integrated
All-GFs F-Score (before LDD Resolution)	79.82	81.12
All-GFs F-Score (after LDD Resolution)	83.79	86.30
Preds-Only F-Score (before LDD Resolution)	70.00	73.45
Preds-Only F-Score (after LDD Resolution)	73.78	78.76

Table 4.25: Evaluation of Parser Output against the DCU 105 gold standard with and without LDD Resolution

Entries also in reference lexicon:	89.89%
Entries not in reference lexicon:	10.11%
Known words:	7.85%
- Known words, known frames:	7.85%
- Known words, unknown frames:	0
Unknown words:	2.32%
- Unknown words, known frames:	2.32%
- Unknown words, unknown frames:	0

Table 4.26: Coverage of Induced Lexicon (WSJ 02-21) on Unseen Data (WSJ 23)(Verbs Only)

4.6.4 Effect of Induced Lexical Resources

As described in the previous section, LDD resolution requires automatically extracted paths and semantic forms along with their associated conditional probabilities. In this section, I report on the effect that resolving LDDs in this way has on the quality of parser output.

Cahill et al. (2004b) report on the improvement in parser performance when LDD resolution is employed using the DCU 105 as a gold standard. The results for a simple PCFG and an A-PCFG in Table 4.25 demonstrate that resolving LDDs using the induced lexical information and functional uncertainty paths improves overall parser performance by 3.97% (all-grammatical-functions) and 3.78% (preds-only) f-score for the pipeline model, and 5.18% (all-grammatical-functions) and 5.31% (preds-only) f-score for the integrated model.

4.7 Coverage and Rate of Accession of the Induced Lexicon

In this section, I examine the rate at which semantic forms are induced by our system.

Entries also in reference lexicon:	58.09%
Entries not in reference lexicon:	41.92%
Known words:	21.21%
- Known words, known frames:	21.21%
- Known words, unknown frames:	0
Unknown words:	20.71%
- Unknown words, known frames:	20.71%
- Unknown words, unknown frames:	0

Table 4.27: Coverage of Induced WSJ Lexicon on Unseen Data (Brown)(Verbs Only)

Entries also in reference lexicon:	65.37%
Entries not in reference lexicon:	34.66%
Known words:	15.63%
- Known words, known frames:	15.60%
- Known words, unknown frames:	0.03
Unknown words:	19.03%
- Unknown words, known frames:	19.03%
- Unknown words, unknown frames:	0

Table 4.28: Coverage of Induced WSJ Lexicon on Unseen Data for Occurrences >1 (Brown)(Verbs Only)

This can be expressed as a measure of the coverage of the induced lexicon on new data. Following experiments carried out by Xia (1999), Hockenmaier et al. (2004) and Miyao et al. (2004), I extract a reference lexicon from Sections 02-21 of the WSJ. I then compare this to a test lexicon from Section 23. Table 4.26 shows the results of the evaluation of the coverage of an induced lexicon for verbs only without including prepositions and particles in the extracted frames. There is a corresponding semantic form in the reference lexicon for 89.89% of the entries in the lexicon induced from Section 23. 10.11% of the entries in the test lexicon did not appear in the reference lexicon. Within this group, I can distinguish between known words, which have an entry in the reference lexicon, and unknown words, which do not exist at all in the reference lexicon. In the same way I make the distinction between known frames and unknown frames. There are, therefore, four different cases in which an entry may not appear in the reference lexicon. Table 4.26 shows that the most common case is that of known verbs occurring with a different, although known, subcategorisation frame (7.85%).

Carroll and Rooth (1998) and Roland and Jurafsky (1998) point out that verbal subcategorisation tendencies vary across linguistic domain. To demonstrate this, I extract a reference lexicon from the entire WSJ section of the Penn-III Treebank and a test lexicon from the parse-annotated Brown subcorpus, the other component of Penn-III. The results of this experiment, shown in Table 4.27, attest a certain amount of domain specificity on the part of the WSJ lexicon. 58.10% of the entries in the Brown lexicon also occur in the WSJ lexicon. Of those which are not contained in the reference lexicon, 21.21% are known words occurring with different but previously seen frames. The other 20.71% are new verbs for which there is no entry in the WSJ lexicon. This 20.71% of the lexicon comprises entries for just 911 verb types many of which are very rare. 755 of these verb types occur only once in the Brown subcorpus. This makes it difficult to have confidence in the reliability of these entries. If I exclude all entries occurring only once in both the reference and test lexicons, the coverage of the WSJ lexicon on the Brown lexicon increases by 7% (Table 4.28). The majority of the unseen entries are made up of new verbs. There is just one unseen frame and 15.63% of the entries are made up of new combinations of known verbs and known frames.

The rate of accession may also be represented graphically. In (Charniak, 1996) and (Krotov et al., 1998), it was observed that treebank grammars (CFGs extracted from treebanks) are very large and grow with the size of the treebank. I was interested in discovering whether the acquisition of lexical material from the same data displayed a similar propensity. Figure 4.7 graphs the rate of induction of semantic form and CFG rule types from Penn-III (the WSJ and parse-annotated Brown Corpus combined). Due to the variation in the size of sections between the Brown and the WSJ, I plotted accession rate against word count. The first part of the graph (up to 1,004,414 words) represents the rate of accession from the WSJ and the final 384,646 words are those of the Brown Corpus. The seven curves represent the following: the acquisition of semantic form types (non-empty) for all syntactic categories with and without specific preposition and particle information, the acquisition of semantic form types (non-empty) for all verbs with and without specific preposition and particle information, the number of lemmas associated with the extracted semantic forms and the acquisition of CFG rule types. The curve

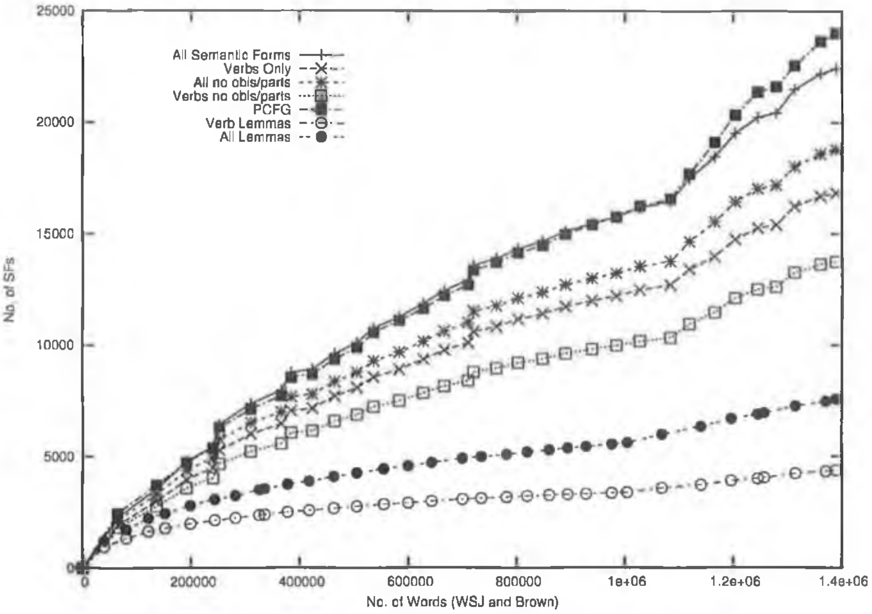


Figure 4.7: Comparison of Accession Rates for Semantic Form and CFG Rule Types for Penn-III (Non-Empty Frames) (WSJ followed by Brown)

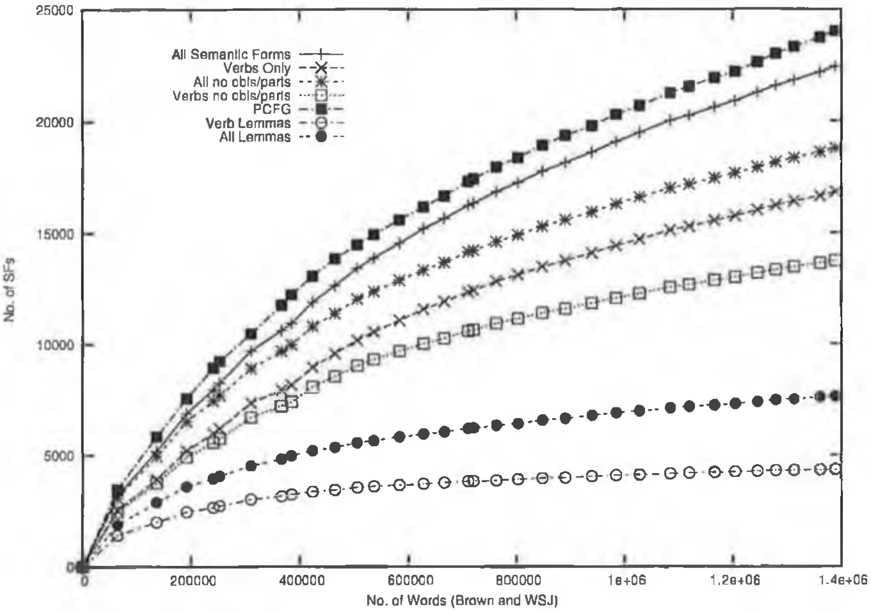


Figure 4.8: Comparison of Accession Rates for Semantic Form and CFG Rule Types for Penn-III (Non-Empty Frames) (Brown followed by WSJ)

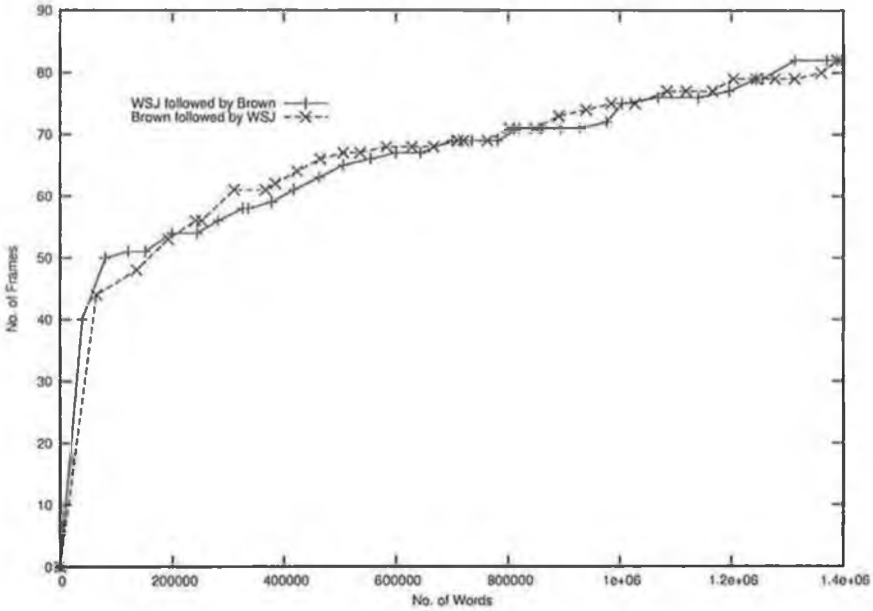


Figure 4.9: Accession Rates for Frame Types (without Prepositions and Particles) for Penn-III

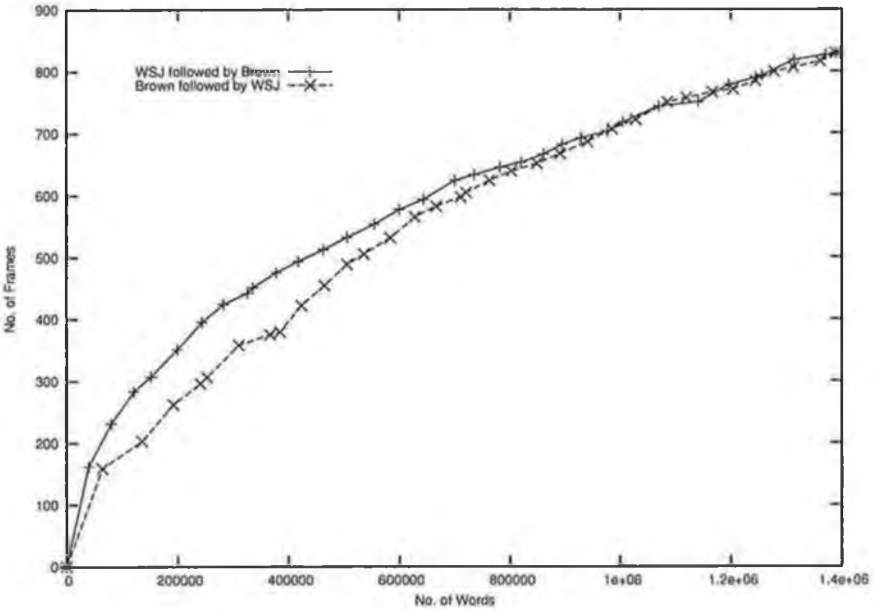


Figure 4.10: Accession Rates for Frame Types (with Prepositions and Particles) for Penn-III

representing the growth in the overall size of the lexicon is similar in shape to that of the PCFG, while the rate of increase in the number of verbal semantic forms (particularly when obliques and particles are excluded) appears to slow more quickly. Figure 4.7 shows the effect of domain diversity from the Brown section in terms of increased growth rates for 1e+06 words upwards. Figure 4.8 depicts the same information but this time extracted from the Brown section first followed by the WSJ. The curves are different but similar trends are represented. This time the effects of domain diversity for the Brown section are discernible by comparing the absolute accession rate for the 0.4e+06 mark between Figures 4.7 and 4.8.

Figure 4.9 shows the result when I abstract away from semantic forms (lemma-frame combinations) to subcategorisation frames and plot their rate of accession. The graph represents the growth rate of frame types for Penn-III (WSJ followed by Brown and Brown followed by WSJ). The curve rises sharply initially but gradually levels, practically flattening out, despite the increase in the number of words. This reflects the information about Section 23 and the Brown subcorpus in Tables 4.26 and 4.27 where I demonstrate that, although new verb+frame combinations occur, all of the frame types in test lexicons have been seen by the lexical extraction program in previous sections. Figure 4.10 shows that including information about prepositions and particles in the frames results in an accession rate which continues to grow, albeit ever more slowly, with the increase in size of the extraction data. This emphasises the advantage of our approach which extracts frames containing such information without the limitation of predefinition.

4.8 Summary

In this chapter, I have applied the methodology presented in Chapter 3 to the entire Penn-III Treebank. I extract a total of 21,005 semantic form types for 4,362 verb lemmas. Including specific prepositions and particles, I extract 1084 frame types. As the lexicon is integrated into a parsing system, it was important to evaluate it in its entirety rather than focusing on a small subset of frequently occurring verbs. To facilitate this, I use two machine-readable lexicons as gold standards rather than a custom-made manually constructed gold standard.

The first of these resources is COMLEX, a machine-readable subcategorisation lexicon. While using COMLEX is clearly less labour-intensive than manually constructing a gold standard, finding a suitable common encoding between it and the induced resource is a non-trivial task. I outline two different mappings and their effect on results. For the second mapping, I evaluate a lexicon induced from the Penn-II (WSJ), a lexicon induced from the parse-annotated Brown Corpus and finally one extracted from the complete Penn-III Treebank (WSJ and Brown combined). I use relative thresholds (1% or 5%) to filter out erroneous frames and vary the level of preposition and particle information in the frames. The largest evaluation is for 3535 verb lemmas, to my knowledge the most extensive evaluation carried out for an automatically induced English lexicon. I achieve an f-score of 65.2% without prepositions and particles, and 35.1% with. The reason for the lower f-score when prepositions and particles are included in the frame can be attributed to overgeneration in the gold-standard lexicon. Incorporating directional prepositions into the induced lexicon following the COMLEX approach increases the f-score from 35.1% to 54.7%. These experiments were carried out for active frames. I also evaluated the induced passive frames in two ways. First, I converted the active frames in the gold standard to passive frames where possible using Lexical Redundancy Rules and compared the induced passive frames to the converted gold-standard frames. Second, I applied Lexical Redundancy Rules in reverse to the induced passive frames converting them to their active equivalents. These were then added to the lexicon which I re-evaluated against COMLEX, this time for 3673 verb lemmas. I achieved similar results to before despite the larger evaluation. I also experimented with absolute rather than relative thresholds. Finally I provide a detailed error analysis of our results against COMLEX. This highlighted some of the drawbacks of using COMLEX as a gold standard as well as pointing to areas of improvement for my extraction system.

The second gold standard I used was the machine-readable version of the OALD. This is quite different to COMLEX as it is not a subcategorisation lexicon. Each verb is associated with a set of patterns which may contain what my system would certainly classify as adjuncts, such as infinitival clauses of purpose. Again finding a common representation for evaluation posed some challenges. I amended our extraction system to include more fine-grained details for clausal arguments in particular. The OALD does not contain information about specific prepositions and particles for OBLs and PARTs, thereby preventing me from evaluating that aspect of my extraction system. The one exception is the *for*-PP and the *to*-PP which alternate with the second object in ditransitive verbs. Frequency thresholds were not adequate to ensure accuracy in frame assignment for these cases so I introduced a diathesis alternation rule into the filtering phase. This improved the f-score by almost 2% to 64.1%. I also include an error analysis section for the OALD evaluation. I found that in particular 7 out of 10 of the transitive fps were correct but did not occur in the gold standard.

I describe a task-based evaluation of the induced lexicon. Following Cahill et al. (2004b), I outline how the automatically acquired semantic forms are used in the LDD resolution component of an probabilistic LFG parsing system. Following theoretical LFG, LDD resolution occurs at f-structure level using functional uncertainty paths and induced semantic forms. In an evaluation against the DCU 105, I demonstrate that incorporating this information into the parsing architecture for the resolution of LDDs results in an improvement of up to 5.31% in parse quality.

I also examine the coverage of the induced lexicon on unseen data. The effect of linguistic domain in lexicon induction is demonstrated when I compute the coverage of a reference lexicon extracted from the WSJ on a test lexicon extracted from the Brown subcorpus.

Chapter 5

Parser-Based Lexical Extraction from the British National Corpus

5.1 Introduction

In this chapter I incorporate the extraction methodology presented in Chapter 3 into a system for the induction of lexical resources from raw text rather than treebank text. The system architecture consists of a preprocessing component (tagger, parser, automatic f-structure annotation, LDD resolution) and the lexical extraction component presented in Chapter 3. To examine the effect of using automatically generated trees rather than gold-standard treebank trees as input to the annotation and extraction components, I carry out an experiment to compare the quality of the lexicon induced from the original treebank trees from Sections 00, 01, 22, 23 and 24 of the Penn-II Treebank, and the quality of the lexicon induced from the corresponding raw strings in those sections using the parser-based architecture. Using a relative threshold of 1% and a variety of parsers and taggers, the experiments show that the treebank-based lexicon is not of statistically significant better quality than even the poorest quality lexicon induced from raw text. This result is both surprising and encouraging as it opens up the possibility of extracting high-quality lexical resources from very large (raw) text corpora. Inducing a lexicon from raw text affords us the opportunity of directly comparing the parser-based extraction system with another, state-of-the-art extraction system, namely that of Korhonen (2002). The experiment shows that the parser and f-structure annotation algorithm-based system performs statistically significantly better than that of Korhonen (2002) at the 95% confidence level using her 65,000 word test corpus, evaluation software and gold standard. In order to examine the robustness and scalability of our system, I extract semantic form lexicons from a 90 million word subcorpus of the British National Corpus (BNC) (Bernard, 2002) and evaluate them against COMLEX and the OALD using the evaluation software described in Chapter 4. I examine the effect of sentence length on lexical quality and coverage, and present the lexical accession rates of this large-scale extraction.

Section 5.2 presents the extraction architecture used to process raw text. Section 5.3 describes the experiments comparing the treebank-based extraction with the parser-based extraction. Section 5.4 outlines the evaluation against the system of Korhonen (2002), presenting results and discussing their significance. Section 5.5 describes the large-scale induction of lexicons from the BNC and their evaluation against COMLEX and the OALD. Section 5.6 examines the effect of sentence length on the accuracy and coverage of the induced lexicons. Finally, Section 5.7 presents the lexical accession rates observed during extraction from the BNC.

5.2 System Architecture

Figure 5.1 outlines the basic procedure for automatically extracting lexical resources from raw text. The model combines and extends work described in (Burke et al., 2004b; Cahill et al., 2004b) and (O'Donovan et al., 2004, 2005a). The text is first tagged and parsed. Due to the modularity of the processing architecture (Figure 5.1), it is possible to experiment with a number of parsers and taggers. The parsers used are Charniak's (2000) maximum entropy inspired parser, Collins' (1999) Models 2 and 3 using beam size 1000 and 10000, and Bikel's (2002) history-based parser. Bikel's parser is retrainable and we use two versions of it: one retains the Penn-II functional tags in training while the other does not. The taggers used are MXPOST (Ratnaparkhi, 1996) and TreeTagger (Schmid, 1994). (Charniak, 2000) is not designed to operate on pre-tagged text but uses its own inbuilt tagger on raw text input. (Bikel, 2002) can optionally be used with its own inbuilt tagger. The resulting parse trees are then passed to the automatic f-structure annotation

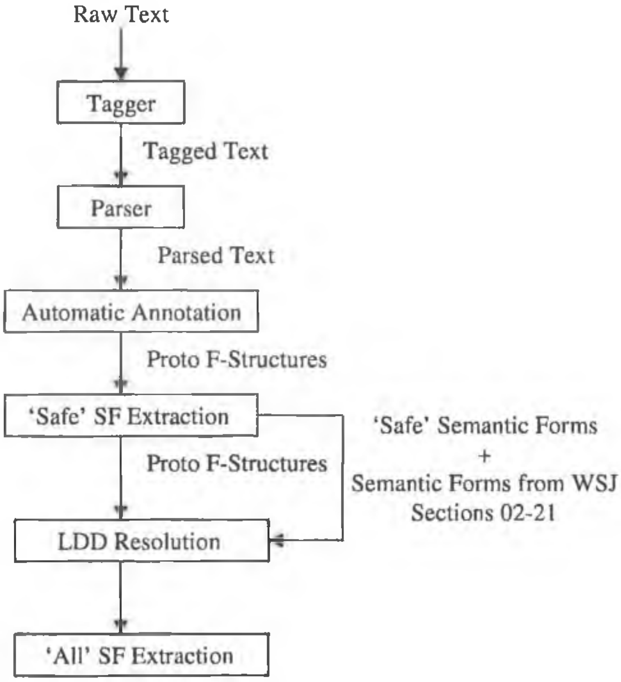


Figure 5.1: Outline of Processing Architecture for the Extraction of Lexical Resources from Raw Text

algorithm described in Chapter 3 which outputs a set of 'proto' f-structures (where LDDs are unresolved).

At this stage in the model it is necessary to extend the LFG parsing architecture architecture in (Cahill et al., 2004b) due to a significant conceptual problem with our parser-based subcategorisation frame extraction. In order to achieve optimal parse results many parsers use subcategorisation frames, but at the same time we are using parsers to 'discover' these very subcategorisation frames. This situation may be viewed as a chicken-and-egg type problem. The parsing component of the subcategorisation frame extraction methodology presented here is based on the pipeline parsing architecture of Cahill et al. (2004b) (see Chapter 4, Figure 4.6). Cahill et al. (2004b) use Penn-II-trained PCFG or history-based lexicalised parsers as the c-structure engines to parse raw text into LFG f-structures. The PCFG parsers are not lexicalised and the subcategorisation information–parser problem does not present itself. The matter is slightly different with the higher performing history-based and lexicalised parsers (Collins, 1999; Charniak, 2000; Bikel, 2002) used in the pipeline architecture. These parsers use subcategorisation information (crudely speaking, in the form of dependencies) learned from the Penn-II training set material (WSJ Sections 02-21) and employ sophisticated back-off methods if dependencies relevant to unseen text are not supported in training material, effectively switching back to unlexicalised PCFG behaviour. In the subcategorisation frame extraction methods presented here, the most significant component where the aforementioned chicken-and-egg problem strikes is in the LDD resolution component of Cahill et al. (2004b). Following the general LFG architecture, in the approach of Cahill et al. (2004b) LDDs are resolved at the level of f-structure, but with finite approximations of functional uncertainty (FU) equations[1] automatically learned from the Penn-II training set (WSJ Sections 02-21), obviating the need for traces (empty productions) and coindexation at c-structure.[2] In order to resolve a (finite approximation of an) FU equation, subcategorisation information is crucial:

- We need to check whether the terminating grammatical function in the FU equation is not already present at the local embedded f-structure reached by the FU path. (If the function is already present, the path cannot be resolved at this level.)
- We need to check whether the local predicate at the relevant level of embedding in the f-structure *subcategorises* for the terminating grammatical function in the FU equation. (If not, the path cannot be resolved at this level.)

LDD resolution is what turns the proto f-structures generated from parser output into proper f-structures. It requires subcategorisation information but at the same time we are using this parsing architecture to 'discover'/extract f-structure based subcategorisation information. In addition, in Chapter 3 I make much of the fact that our treebank-based subcategorisation frame extraction method is one of the few methods where the frames extracted fully reflect the effects of LDDs as encoded in the underlying treebank tree data-structures. How can we achieve this in the parser-based extraction method, knowing that to resolve LDDs requires subcategorisation information we are about to discover using parser-based extraction?

[1] Functional uncertainty equations are regular expressions over paths in f-structure relating filler-gap material in terms of re-entrancies at f-structure (Cf. Chapter 4).

[2] Unlike the original Penn-II treebank trees, standard PCFG and history-based parsing technology does not capture LDDs. The exception is Collins' Model 3 which captures a limited number of LDDs in relative clause constructions.

The key to solving this conundrum lies in the observation that the proto f-structures generated for parser output already contain a subset of reliable and locally complete f-structure levels and hence semantic forms extracted from those levels are reliable and complete. How can we identify levels of embedding in proto f-structures that are reliable and complete? The answer is surprisingly simple: reliable and complete levels of f-structure within proto f-structures are those levels of f-structure which:

- do not feature an (unresolved) LDD indicator such as TOPIC, TOPICREL, FOCUS, etc.
- and are not embedded within a level of f-structure that features an (unresolved) LDD indicator such as TOPIC, TOPICREL, FOCUS, etc. with the exception where the embedding is via an LDD indicator.

For the simple example f-structure in Figure 5.2, I extract a lexical entry for the verb `be` (`be([subj,xcomp]`), but not for the verb `say` as `say` is embedded at a level of f-structure which contains an unresolved LDD indicator. I then combine this subset of reliable semantic forms from newly parsed unseen text with those extracted from the original treebank trees in Sections 02-21 of the WSJ[3] (the training set for the statistical parsers) and recalculate the probability associated with each semantic form. This resource is then exploited by the LDD resolution component of (Cahill et al., 2004b) to resolve any LDDs in the f-structures for unseen text (Figure 5.1). Finally, the lexical extraction step is rerun on the resulting set of 'proper' f-structures, this time extracting all semantic forms. For the resolved f-structure in Figure 5.3 I now extract two semantic forms: `be([subj,xcomp])` and `say([subj,comp])`.

The quality of the f-structures produced by the tagging, parsing and automatic annotation steps is of crucial importance to the quality of the induced lexical resource. Cahill et al. (2005a) show that statistical grammars enriched with automatic f-structure annotations, as used in the architecture presented here, significantly outperform existing hand-crafted, deep wide-coverage grammars. The best system achieves an f-score of 83.17% against the PARC-700 Dependency Bank, a 2.62% improvement on the most re-

[3] Recall that the original treebank trees fully encode LDDs in terms of empty productions (gaps) coindexed with lexicalised material (fillers), and the f-structure annotation algorithm translates those coindexations into corresponding re-entrancies at f-structure, fully reflecting the LDDs in the source treebank tree data-structures

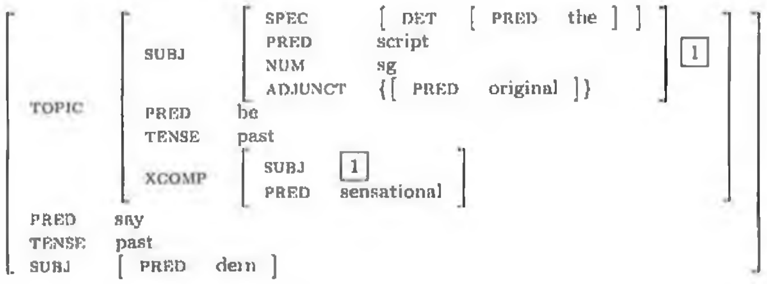


Figure 5.2: Proto F-Structure for the BNC String The original script was sensational, Dern said. before LDD Resolution

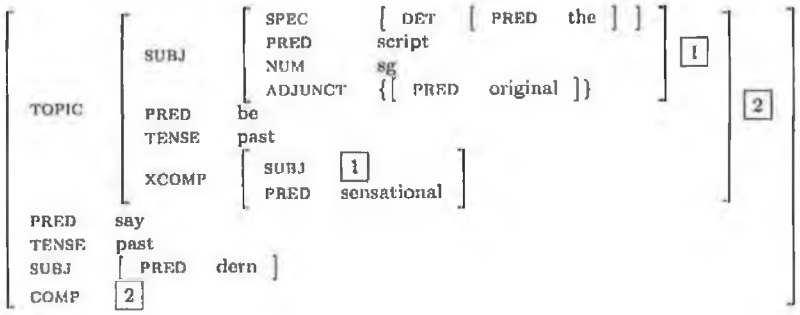


Figure 5.3: Proper F-Structure for the BNC String The original script was sensational, Dern said. after LDD Resolution

Parser	PARC-700	CBS-500
Bikel retrained	83.17	80.22
Collins M2	80.53	76.13
Collins M3	80.50	75.83
Charniak	82.05	78.33
XLE	80.55	
RASP		76.57

Table 5.1: Dependency-Based Parser Evaluation (F-Scores) against the PARC-700 and CBS-500

cent results for the hand-crafted LFG grammar and XLE parsing system of Riezler et al. (2002), and an f-score of 80.22% against the CBS-500 Dependency Bank (Carroll et al., 1998a), a 3.65% improvement over the hand-crafted RASP grammar and parsing system of Carroll and Briscoe (2002). The results are summarised in Table 5.1. The system using the retrained Bikel parser as c-structure engine with the f-structure annotation algorithm performs best against both gold standards.

5.3 Treebank- versus Parser-Based Lexicon Extraction

Using a tagger and a parser to automatically produce trees for input to the f-structure annotation and lexical extraction components is likely to introduce a greater margin of error than using hand-corrected treebank trees. In order to measure the effect of parser-based lexical extraction on the resulting lexicon, I compare the lexical resources extracted from the original treebank trees from Sections 00, 01, 22, 23 and 24 of the WSJ section of the Penn-II Treebank with the results of parser-based extraction from the raw text from the same sections of Penn-II. Sections 00, 01, 22, 23 and 24 contain a total of 9375 sentences with 223708 word tokens. In each case, the parsers are trained on WSJ Sections 02-21. The extracted lexicons are evaluated against COMLEX. I use combinations of the parsers and taggers described in Section 5.2. Except when using Charniak's (2000) parser[4], we also use the hand-corrected tagged version of the WSJ text as input to the extraction system to examine the margin of error introduced by the tagging step.

I carried out two evaluations of the induced lexical resources against COMLEX fol-

[4] Recall that (Charniak, 2000) is not designed to take tagged input.

Parser	Tagger	# Parses	#Verbs	Precision	Recall	F-Score
Charniak	Own	9370	1384	64.06	45.54	53.24
Collins M2 (1000)	Gold	9364	1415	65.02	44.87	53.10
Collins M2 (1000)	TreeTagger	9250	1404	63.44	45.09	52.71
Collins M2 (1000)	MXPost	9364	1390	65.10	44.97	53.19
Collins M2 (10000)	Gold	9358	1412	65.31	45.06	53.33
Collins M2 (10000)	TreeTagger	9245	1404	63.62	45.22	52.87
Collins M2 (10000)	MXPost	9356	1386	65.33	45.13	53.38
Collins M3 (1000)	Gold	9356	1413	65.23	45.01	53.27
Collins M3 (1000)	TreeTagger	9245	1403	63.38	45.06	52.67
Collins M3 (1000)	MXPost	9356	1386	65.22	45.11	53.33
Collins M3 (10000)	Gold	9349	1412	65.28	45.06	53.32
Collins M3 (10000)	TreeTagger	9238	1403	63.63	45.17	52.84
Collins M3 (10000)	MXPost	9348	1383	65.26	45.15	53.38
Bikel	Gold	9368	1414	65.29	44.65	53.03
Bikel	TreeTagger	9252	1409	63.61	44.93	52.67
Bikel	MXPost	9364	1395	65.38	44.76	53.14
Bikel	Own	9369	1380	65.28	44.86	53.18
Bikel Retrained	Gold	9367	1407	80.30	43.30	56.27
Bikel Retrained	TreeTagger	9258	1401	77.52	43.74	55.92
Bikel Retrained	MXPost	9363	1392	80.36	43.38	56.34
Bikel Retrained	Own	9368	1368	79.76	43.38	56.19
Treebank Trees		9375	1449	86.06	41.78	56.25

Table 5.2: Results with All Verbs against COMLEX (Relative Threshold 1%)

lowing COMLEX-LFG Mapping 2 outlined in Chapter 4. In both cases I filtered the automatically acquired resources using a relative threshold of 1%. Table 5.2 contains the results of evaluating the entire induced lexicon for each tagger/parser combination as well as that for the gold standard treebank trees. The second column contains the number of sentences (out of 9375) successfully parsed in each case. All parsers achieve over 98.5% coverage. Due to variation in lexicon size (fourth column in Table 5.2), it is difficult to compare the quality of the lexicons here. It is interesting to observe this variance in size, however. The largest lexicon is that induced from the gold-standard treebank trees (1449 verb lemmas). The next largest is that induced using Collins Model 2 with a beam size of 1000 and the gold-standard Penn-II tags (1415 verb lemmas). For the parsers used with MXPost, TreeTagger and the gold standard tags (both Collins' Models and Bikel), the lexicon induced using the gold standard tags is consistently larger than that using either of the taggers. Using TreeTagger seems to result in a slightly larger lexicon than using MXPost. This is surprising, as generally the parser coverage is lower when TreeTagger is used. There are strings in these WSJ sections, however, which do not contain a verb (e.g. lists of share prices and interest rates). Failure to parse these will not have a detrimental effect on the number of verb lemmas in the lexicon. Using the Bikel parser retrained to retain Penn-II functional labels results in a slightly smaller lexicon for each tagger than for the original Bikel parser. An obvious pattern which emerges in the evaluation scores is that the lexicons extracted using the Collins, Bikel and Charniak parsers as c-structure engines result in lower precision (by between 10 and 20%) but higher recall than the lexicons extracted both from the treebank and using Bikel Retrained. The main reason for this is the way in which obliques are identified. For Bikel Retrained and the original treebank trees, I use the conservative but precise -CLR tag as described in Chapter 3. For the other parsers (which produce 'raw' trees without Penn-II functional tags), I apply the methodology developed for the Brown Corpus subsection of Penn-III (cf. Section 4.2): the PP sister of a V daughter of a VP node is annotated as an oblique. This methodology is less precise but also less conservative than the -CLR method which is reflected in the results. A relative threshold of 1% is possibly too low to filter out the incorrectly extracted obliques. In general, the recall scores are low due to the limited size of the data set (just over 200,000

add	become	begin	buy	close	come
continue	end	expect	fall	get	give
go	help	hold	include	increase	make
pay	report	rise	see	sell	take
think	use	want			

Table 5.3: The 27 Test Verbs used in the WSJ Evaluation

words). Surprisingly, recall scores are higher for the parsers than for the original treebank trees suggesting that some parser errors actually result in frames attested in COMLEX.

For the second evaluation, I used a subset of 27 verbs with corpus frequency between 100 and 500 which all of the induced lexicons in Table 5.2 have in common. The verbs are shown in Table 5.3. A controlled evaluation like this allows for a more telling comparison between the quality of the various induced lexicons. The results are shown in Table 5.4. The lexicon induced from the treebank trees achieves a higher f-score (71.07%) than any of the parser-based lexicons. The best of the parser-based lexicons (69.00% f-score) is induced using the retrained Bikel parser and the gold standard tags. The f-scores achieved using the retrained Bikel parser with all taggers are higher than the f-scores achieved using any other parser. The next highest score is that for Collins Model 3 with a beam size of 10000 and TreeTagger (67.50%). Using the gold-standard treebank tags does not seem to give an advantage over the automatic tagging. The only case where there is an advantage is for the retrained Bikel parser (69.00% f-score versus 68.80%). The difference in precision and recall patterns between Bikel Retrained and the treebank trees, and the other parser-based methods observed in the discussion of Table 5.2 is even more obvious for this smaller evaluation.

5.3.1 Significance Testing

While using the treebank trees seems to produce a lexicon of better quality, when I examine the difference more closely I find that it is not statistically significantly better than the best performing parser and tagger combination (Bikel Retrained with its own tagger, p-value=0.24) nor even statistically significantly better than the worst performing parser and tagger combination (Collins Model 2 with beam size 1000 and MXPost, p-value=0.06)

Parser	**Tagger**	**Precision**	**Recall**	**F-Score**
Charniak	Own	61.24	73.83	66.95
Collins M2 (1000)	Gold	59.16	72.43	65.13
Collins M2 (1000)	TreeTagger	58.87	72.90	65.14
Collins M2 (1000)	MXPost	59.46	71.96	65.12
Collins M2 (10000)	Gold	59.54	72.90	65.55
Collins M2 (10000)	TreeTagger	60.46	74.30	66.67
Collins M2 (10000)	MXPost	60.08	72.43	65.68
Collins M3 (1000)	Gold	60.00	74.30	66.39
Collins M3 (1000)	TreeTagger	60.38	74.77	66.81
Collins M3 (1000)	MXPost	59.62	73.83	65.97
Collins M3 (10000)	Gold	60.00	74.30	66.39
Collins M3 (10000)	TreeTagger	60.90	75.70	67.50
Collins M3 (10000)	MXPost	59.85	73.83	66.11
Bikel	Gold	59.02	73.36	65.42
Bikel	TreeTagger	59.92	73.36	65.97
Bikel	MXPost	59.70	73.36	65.83
Bikel	Own	59.62	73.83	65.97
Bikel Retrained	Gold	81.53	59.81	69.00
Bikel Retrained	TreeTagger	78.18	60.28	68.07
Bikel Retrained	MXPost	80.00	59.81	68.45
Bikel Retrained	Own	80.12	60.28	68.80
Treebank Trees		86.58	60.28	71.07

Table 5.4: Results with 27 Frequently Occurring Verbs against COMLEX (Relative Threshold 1%)

at the 95% confidence level (with the p-value less than 0.05), using the Approximate Randomization Test (Noreen, 1989). These results are surprising and encouraging as they open up the possibility of extracting high-quality lexical resources from very large (raw) text corpora. If I do not use a relative threshold to filter the results this has a more damaging effect on the parser-based performances than the treebank-based performance (cf. Table 5.5). The one exception is Bikel Retrained with gold-standard tags which outperforms the treebank-based extraction. Aside from the Bikel Retrained induced lexicons, there is now a statistically significant difference between the f-score for the treebank-based resource and the parser-based resources. It is evident that some statistical filtering is required for the noisier parser-based approach. While the statistical parsers used here will perform best on WSJ data as it most resembles their training data, I believe that the results of this experiment justify the use of the parser-based architecture to automatically extract lexical resources from raw text.

At this stage, however, it is very hard to know whether these results are truly conclusive. One limiting factor may be the test set size (223708 word tokens from WSJ Sections 00, 01, 22, 23 and 24) for both experiments reported here. In future work, we will explore a cross validation experiment, parsing the entire Penn-II data by moving a 5 Section window through Penn-II and training with the remaining sections to extract and compare complete parser-based and original treebank tree-based lexical resources against COMLEX and the OALD. Furthermore, discussed above, there may be a masking effect where mis-parses accidentally result in frames attested in the gold standard. This possibility can only be ruled out by detailed manual error analysis.

5.3.2 Assessing the Impact of LDD Resolution

In order to assess the impact of LDD resolution in the extraction architecture, I carry out an experiment using the best performing parser (Bikel Retrained) with the gold-standard treebank tags on Sections 00, 01, 22, 23 and 24 of the WSJ.

I carry out three evaluations against COMLEX using the 27 frequently occurring test verbs listed in Table 5.3 and no relative threshold. The first evaluation (Experiment 1) is for the set of all semantic forms extracted from 'proto' f-structures where LDD resolution

Parser	Tagger	Precision	Recall	F-Score
Charniak	Own	49.18	84.58	62.20
Collins M2 (1000)	Gold	48.11	83.18	60.96
Collins M2 (1000)	TreeTagger	47.87	84.11	61.02
Collins M2 (1000)	MXPost	47.71	82.71	60.51
Collins M2 (10000)	Gold	48.49	82.71	61.14
Collins M2 (10000)	TreeTagger	48.53	84.58	61.67
Collins M2 (10000)	MXPost	48.09	82.24	60.69
Collins M3 (1000)	Gold	48.36	82.71	61.03
Collins M3 (1000)	TreeTagger	47.61	83.64	60.68
Collins M3 (1000)	MXPost	47.96	82.24	60.59
Collins M3 (10000)	Gold	48.63	82.71	61.25
Collins M3 (10000)	TreeTagger	48.26	84.11	61.33
Collins M3 (10000)	MXPost	48.22	82.24	60.79
Bikel	Gold	48.48	82.24	61.01
Bikel	TreeTagger	47.34	83.18	60.34
Bikel	MXPost	48.48	82.24	61.01
Bikel	Own	48.90	83.18	61.59
Bikel Retrained	Gold	71.43	74.77	73.06
Bikel Retrained	TreeTagger	68.40	73.83	71.01
Bikel Retrained	MXPost	69.60	73.83	69.60
Bikel Retrained	Own	69.74	74.30	71.95
Treebank Trees		75.00	70.09	72.46

Table 5.5: Results with 27 Frequently Occurring Verbs against COMLEX (No Filtering)

	Precision	Recall	F-Score
Experiment 1	71.49	73.83	72.64
Experiment 2	73.11	72.43	72.77
Experiment 3	71.43	74.77	73.06

Table 5.6: Evaluation of Induced Semantic Forms without and without LDD Resolution

is not included in the extraction architecture. Experiment 2 is an evaluation of the 'safe' semantic forms in 'proto' f-structures before they are combined with the semantic forms from Sections 02-21 for LDD resolution in Experiment 3 which measures the quality of all extracted semantic forms after LDD resolution. In Experiment 1 I extract 3439 semantic forms, in Experiment 2 I extract 3135 'safe' semantic forms and in Experiment 3 3443 semantic forms. The reason the number of semantic forms for Experiment 3 is slightly higher than that for Experiment 1 is that our system disregards empty frames which are likely to occur in Experiment 1 if the moved element in a sub-f-structure is a subject. The results are presented in Table 5.6. As expected the highest f-score (73.06%) is recorded for Experiment 3 (with LDD resolution) and the lowest (72.64%) for Experiment 1 (without LDD resolution). The most precise lexicon is (unsurprisingly) the 'safe' semantic forms (73.11%) but this lexicon scores lowest on recall (72.43%). Precision for Experiment 3 is slightly lower than that for Experiment 1 (0.06%) but Experiment 3 has very high recall (74.77% against 73.83% and 72.43%).

Although this is a small evaluation, the results are encouraging and confirm expectations regarding the effect of incorporating our novel approach to LDD resolution in the extraction system.

5.4 Evaluation against Korhonen (2002)

In order to assess the quality of the parser-based lexical extraction system it is important to compare the system with other state-of-the-art parser-based extraction systems. I use Anna Korhonen's publicly available evaluation resources[5] in order to compare the system with that of Korhonen (2002). The resources include a 65,000 word subcorpus from the

[5] These evaluation resources were downloaded from http://www.cl.cam.ac.uk/users/alk23/subcat/subcat.html

add	agree	attach	bring	carry	carve
chop	cling	clip	communicate	cut	drag
fly	give	lend	lock	marry	meet
mix	move	offer	provide	push	sail
send	slice	swing	travel	visit	

Table 5.7: The 29 Test Verbs used in the Korhonen Evaluation

BNC, a manually constructed gold standard (for 30 test verbs), evaluation software and the best results achieved by Korhonen (2002). As the corpus data for one of the verbs (`supply`) are presently unavailable, I compare the systems for 29 verbs. The Korhonen test verbs (listed in Table 5.7) occur on average 1000 times in the constructed subcorpus and were chosen as they display a variety of subcategorisation patterns. As Bikel's (2002) parser retrained to retain Penn-II functional tags outperformed the other parsers used in the WSJ experiments described in Section 5.3 (cf. Tables 5.2 and 5.4), I use it in combination with Schmid's (1996) TreeTagger. The experiments in Section 5.3 also demonstrated that this parser/tagger combination perform best with a low relative threshold so I set an empirically established threshold of 0.6% to filter the output of the system. The parser and tagger are trained for use on American rather than British English. For this reason, prior to inputting the BNC text of the Korhonen test corpus to the extraction system, I first automatically convert the British spelling to American spelling using a tool called `varcon` designed for this purpose.[6]

5.4.1 Mapping

The subcategorisation frames extracted by Korhonen (2002) differ considerably in format to ours. It was necessary, therefore, to convert the GF-based frames to the corresponding frames used in the Korhonen (2002) gold standard. This involves adapting our automatic extraction system to automatically provide additional information to support the mapping to the Korhonen gold standard frames. Korhonen (2002) uses 43 frames in the gold standard.[7] I map to 33 of these frames. For the remaining 10 frames, I score zero in the

[6] `varcon` can be downloaded from http://wordlist.sourceforge.net/

[7] The system of Korhonen (2002) is based on that of Briscoe and Carroll (1997) which uses 163 predefined frames in total. However, our aim is to evaluate against the provided gold standard and not to replicate the Briscoe and Carroll (1997) system. For this reason, I confine the mapping to the gold standard frames.

evaluation.[8] There are two main reasons for not mapping to the remaining 10 frames. The first of these is related to the control information encoded in the gold standard frames. While our system can differentiate between subject and object control, it does not yet distinguish subject control from raising and so cannot extract an equivalent to the Korhonen (2002) gold standard frame 111. The second reason for not mapping to all gold standard frames relates to the encoding of surface order. Our frames represent an alphabetically ordered list of subcategorised for grammatical functions and do not depict surface order. In LFG, linearisation is done by the annotated c-structure rules, not by the subcategorisation frames (semantic forms). Therefore, I do not differentiate between frames 78 and 119 (both are [obl:*pval*,part:*adval*]) or between frames 77 and 120 (both are [obj,obl:*pval*,part:*adval*]).

For the most part, the mapping involved automatically extracting extra information about the objects of prepositional phrases and the conjunction introducing clauses. For example, instead of [obl:*pval*,obl2:*pval*][9], our modified system extracts the more detailed [obl:*pval*,obl2(pwh)] which is mapped to the Korhonen gold standard frame 91 (PP-P-WH-S). Similarly, [comp] becomes [comp(that)] which maps to the gold standard frame 109 (THAT-S). To give a simple example, from the f-structure shown in Figure 5.4 the frame insist([subj,obl(ping)]) rather than the usual insist([subj,obl:on]) is extracted. I induce the necessary information from the object of the oblique. In this case, the crucial detail is that the verb *keep* is a present participle (PARTICIPLE pres), recorded as ping in the frame. As subject information is not included in the gold-standard frames, subj is removed from the frames. The full details of the mapping are shown in Table 5.8. The first column contains the original frame format, the second shows the automatically enriched frame, if applicable, and the third shows the equivalent Korhonen (2002) gold-standard frame.

[8] As I do not assign a score of 0% to the output of Korhonen's system for these frames, the evaluation is potentially biased against our system.

[9] To comply with the Korhonen gold standard, the prepositions associated with oblique arguments are generally replaced with a *pval* dummy value, except for cases where a particular preposition is specified as part of the gold standard frame (e.g. frames 31 and 29).

$$\left[\begin{array}{ll}
\textsc{subj} & \left[\begin{array}{ll}
\textsc{adjunct} & \left\{\left[\begin{array}{ll}\textsc{pred} & \text{mr}\\ \textsc{num} & \text{sg}\\ \textsc{pers} & 3\end{array}\right]\right\}\\
\textsc{pred} & \text{kaifu}\\
\textsc{num} & \text{sg}\\
\textsc{pers} & 3
\end{array}\right]\boxed{1}\\
\textsc{pred} & \text{have}\\
\textsc{tense} & \text{pres}\\
\textsc{cat} & \text{v}\\
\textsc{xcomp} & \left[\begin{array}{ll}
\textsc{subj} & \boxed{1}\\
\textsc{pred} & \text{insist}\\
\textsc{tense} & \text{past}\\
\textsc{cat} & \text{v}\\
\textsc{ppart} & +\\
\textsc{obl} & \left[\begin{array}{ll}
\textsc{pform} & \text{on}\\
\textsc{obj} & \left[\begin{array}{ll}
\textsc{pred} & \text{keep}\\
\textsc{participle} & \text{pres}\\
\textsc{cat} & \text{v}\\
\textsc{xcomp} & \left[\begin{array}{ll}\textsc{subj} & \boxed{2}\\ \textsc{pred} & \text{clean}\\ \textsc{cat} & \text{jj}\end{array}\right]\\
\textsc{obj} & \left[\begin{array}{ll}\textsc{pred} & \text{pro}\\ \textsc{pron\_form} & \text{it}\end{array}\right]\boxed{2}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

Figure 5.4: F-Structure from which `insist([subj,obl(ping)])` is extracted from the Sentence: `Mr. Kaifu has insisted on keeping it clean.`

Basic Frame	Enriched Frame	Gold Frame
xcomp	xcomp(jj)	1
xcomp	xcomp(rb)	3, 160
subj	subj	22
subj	subj(pl)	23
obj	obj	24
obj,xcomp	obj,xcomp(jj)	25
obj,xcomp	obj,xcomp(rb)	27
obj,obl:as	obj,obl:as	29
obj,obl:for	obj,obl:for	31
obj,xcomp	obj,xcomp(ing)	35
obj,obj2	obj,obj2	37
obj,obl:pval	obj,obl:pval	49
obj,xcomp	obj,xcomp(inf)	53
obj,obl:to	obj,obl:to	56
obl:pval	obl(ping)	63
comp	comp(ping)	63
obl:pval	obl(pposing)	69
obl:pval	obl(pwh)	70
part:adval	part:adval	74
obj,part:adval	obj,part:adval	76
obj,obl:pval,part:adval	obj,obl:pval,part:adval	77
obl:pval,part:adval	obl:pval,part:adval	78
obl:pval	obl:pval	87
obl:pval,obl2:pval	obl:pval,obl2(pwh)	91
obl:pval,obl2:pval	obl:pval,obl2:pval	95
comp,obl:pval	comp(that),obl:pval	97
comp,obl:pval	comp(what),obl:pval	101
comp	comp(that)	109
xcomp	xcomp(inf)	112
comp	comp(what)	114
obj,obl:pval,obl2:pval	obj,obl:pval,obl2:pval	122
xcomp,part:adval	xcomp(inf),part:adval	150
obj,comp	obj,comp(finf)	157

Table 5.8: Mapping from our Frames to the Korhonen (2002) Gold Standard Frames

5.4.2 Addition of Passive and Diathesis Alternation Rules

As previously mentioned, I extract frames from passive constructions and mark this in the lexicon. In order to extend the coverage of our system, I introduce a number of heuristics to convert the passive frames to their active equivalents. For example, the passive frame `order([subj,xcomp],p)` is rewritten as `order([obj,xcomp])`.

In general, the Korhonen gold standard does not contain information about the specific preposition used in a subcategorised-for PP. Therefore I conflate the specific prepositions in our extracted frames to the general *pval* variable. There are two exceptions in the gold standard however: `[obj,obl:to]` and `[obj,obl:for]`. These frames refer to a special case where ditransitive verbs such as *give* can express the second object in an alternative way. The alternation from `give[obj,obj2]` to `give[obj,obl:to]` is an example of the dative alternation. The alternation from `get[obj,obj2]` to `get[obj,obl:for]` is an example of the benefactive alternation. To deal with this I employ the alternation rules described in Section 4.5.2 for evaluating against the OALD.

5.4.3 Evaluation and Significance Testing

I evaluate the induced lexicon using Korhonen's software. The results are shown in Table 5.9. Without applying the alternation rules described in Section 6.4.1, I report an f-score of 74.91%. Including these rules, I achieve an f-score of 76.16% against the 71.46% f-score produced by replicating the best results of Korhonen (2002) for the 29 test verbs. Interestingly, the alternation rules affect precision rather than recall. This is probably due to the `[obj,obl:to]` and `[obj,obl:for]` frames included in the induced lexicon which should have been mapped to `[obj,obl:`*pval*`]`. I establish the significance of this difference using the Approximate Randomization Test (Noreen, 1989). Comparing the output of our system against that of Korhonen (2002), I get a p-value of 0.014. This is less than 0.05, the smallest fixed level at which the null hypothesis can be rejected, implying that our system significantly outperforms that of (Korhonen, 2002) at the 95% confidence level.

	Precision	Recall	F-Score
Korhonen (2002)	75.90	67.50	71.46
O'Donovan (2006) without Alternation Rules	76.30	73.57	74.91
O'Donovan (2006) with Alternation Rules	78.93	73.57	76.16

Table 5.9: Results of the Evaluation of the DCU extraction system (O'Donovan, 2006) against the Korhonen(2002) Gold Standard

5.4.4 Discussion

Given the difficulty inherent in automatically mapping to a gold-standard format as discussed in Chapter 4 (in general such mappings are lossy: they both over- and under-generalise, i.e. map material they are not supposed to map and miss material they are supposed to map), the results presented in Table 5.9 are very encouraging. The gold standard is provided by frames extracted by the system of Korhonen (2002). Even though at present, I am unable to map to ten gold standard frames for reasons discussed above and suffer a zero score for those in the evaluation, we still outperform the system of Korhonen (2002).

As I am using the evaluation software provided by Korhonen, mapping occurs prior to statistical filtering. For each verb the software takes as input a set of frames complying in format with the gold standard along with their absolute frequencies. This means that only the absolute frequencies of a reduced number of frames which I know are correct (i.e. are contained in the gold standard) are used in the calculation of the conditional probability of a verb–frame combination. Any other frames are rejected and not included in the probability computation. In total, the mapping rejects 59 frames, the majority of which are due to cases where a basic COMP or XCOMP was not automatically augmented to allow for mapping to the gold-standard frames. This biases the performance of our system in a similar way to the approaches of, for example, Brent (1993); Manning (1993); Ushioda et al. (1993); Briscoe and Carroll (1997); Carroll and Rooth (1998); Schulte im Walde (2002a) where frames are effectively filtered through predefinition.

Korhonen et al. (2000) show that using thresholds on relative frequencies outperforms both the Binomial Hypothesis Test and the Binomial Log Likelihood Ratio as a statistical filtering step in lexical extraction. It is, therefore, not surprising that our simple filtering step works quite well. It is interesting, however, that the use of simple statistical filtering results in a better quality output than the use of sophisticated back-off techniques based on verb classes employed by Korhonen (2002). As a conjecture, the performance difference is probably due to the parser and f-structure annotation quality.

5.5 Large-Scale Lexical Extraction from the BNC

The British National Corpus (Bernard, 2002) is a general, synchronic, monolingual (British English) corpus. It contains samples of transcribed spoken and written language. For the purpose of the experiments described here, we are only interested in the written component which comprises 90% of the corpus (approximately 90 million words). The textual component displays domain variation in that 75% is taken from informative writing with the remainder sampled from imaginative writing. As regards medium, 60% of the written texts are sourced from books, 25% from periodicals, between 5 and 10% from miscellaneous published material (e.g. brochures), between 5 and 10% from unpublished material (e.g. letters and essays) and less than 5% from material scripted to be spoken (e.g. speeches and plays). Exploiting the SGML mark-up used in the BNC, I extract an untagged subset of the written component with no limit on sentence length and which excludes list items, captions, labels and quotes[10] (but does include direct and indirect speech). This subset is 91,580,041 word tokens (including punctuation) in size. As for the Korhonen (2002) test corpus, I first convert the British spelling of the BNC to American spelling using `varcon`.

Based on the results obtained in the small-scale evaluation described in Section 5.3 and for reasons of computational efficiency, the parsers and taggers used for this large-scale extraction are (Charniak, 2000) and (Collins, 1999) Model 2 beam size 1000 with TreeTagger together with the f-structure annotation algorithm and the modified LDD resolution architecture as introduced in Section 5.2 (cf. Figure 5.1).[11] To control the quality of the frames, I only accept those containing three grammatical functions or less. The experiments in Section 5.3 demonstrated that for subcategorisation frame extraction

[10] Quotes are literary quotes such as excerpts from poems.

[11] Parsing experiments using Bikel's parser retrained to retain Penn-II functional tags did not finish in time to be included here.

	Collins	Charniak
Sem. Form Types > 1	97937	56884
Active	68203	39539
Passive	29734	17345

Table 5.10: Number of Semantic Form Types Extracted from the BNC (without preposition/particle, occurring >1)

these parser/tagger combinations work best with a higher relative threshold. For the COMLEX (Section 5.5.2) and OALD (Section 5.5.3) evaluations I use a relative threshold of 7.5% to filter erroneous frame to verb assignment by the extraction system. It was also established in Section 5.3 that frames containing obliques in particular are over-generated when these parsers and tagger are incorporated into the system. For this reason, I set an increased relative threshold (15%) to filter frames containing oblique arguments.

	Without Prep/Part		**With Prep/Part**	
	Collins	Charniak	Collins	Charniak
# Frame Types	55	50	3702	2416
# Singletons	0	1	1435	734
# Twice Occurring	0	0	349	245
# Occurring max. 5	2	1	2195	1316
# Occurring > 5	53	49	1507	1100

Table 5.11: Number of Frame Types extracted from the BNC

	Collins	Charniak
# Lemma Types	30026	17173
# Singletons	10348	4957
# Twice Occurring	5207	2402
# Occurring max. 5	20584	9456
# Occurring > 5	9442	7717

Table 5.12: Number of Lemma Types (used actively and passively) extracted from the BNC

5.5.1 Scale of Extracted Lexical Resources

Both Charniak (2000) and Collins (1999) achieve over 99% parser coverage on the test corpus. I extract 97,937 semantic form types (excluding singletons) with Collins and 56,884 with Charniak (cf. Table 5.10). Of the Collins semantic form types, 68,203 are

	Collins	Charniak
Avg. Sem. Form Types	5.6	6.4

Table 5.13: Average Number of Semantic Form Types per Verb extracted from the BNC using Collins and Charniak

active and 29,734 are passive. The ratio is similar for Charniak: 39,539 to 17,345. The number of frame types extracted using each parser is summarised in Table 5.11. As previously mentioned, I limit the frames to only contain three elements or less. Without parameterising for prepositions and particles, I extract 55 frame types using Collins and 50 using Charniak. Setting an absolute threshold of 5 on these frames results in a set of 53 for Collins and 49 for Charniak. When I parameterise for specific prepositions and particles, I extract 3,702 frame types using Collins and 2,416 using Charniak. The difference in these two figures is reduced when I use an absolute threshold of 5: 1,507 for Collins and 1,100 for Charniak. Table 5.12 shows the number of lemma types (active and passive) extracted by system. Without using any filtering, entries for 30,026 lemma types are extracted using Collins and 17,173 using Charniak. Again setting a threshold on lemma frequency reduces the discrepancy between these two figures. 9,442 lemmas occur more than 5 times in the Collins lexicon, with 7717 in the Charniak lexicon. Without using any filtering, I extract an average of 5.6 semantic forms per verb lemma using Collins (1999) and 6.4 semantic forms per verb lemma using Charniak (2000) (cf. Table 5.13). In general, using Collins as part of the lexical extraction architecture, more semantic form, frame and lemma types are extracted. This may primarily be attributed to differences in tagging. I use TreeTagger with Collins while Charniak works with its own inbuilt tagger.

5.5.2 COMLEX Evaluation

For evaluation against COMLEX I only consider active semantic forms for verb lemmas occurring a minimum of five times and only those occurring with frames of three elements or less. Applying these restrictions to the induced lexicons results in a lexicon for 6924 lemmas (Collins) and 7034 (Charniak). Figures 5.5 (Collins and COMLEX) and 5.6 (Charniak and COMLEX) illustrate the overlap between the induced resources and COMLEX. The larger evaluation is for the Collins lexicon (4703 lemmas). In each case

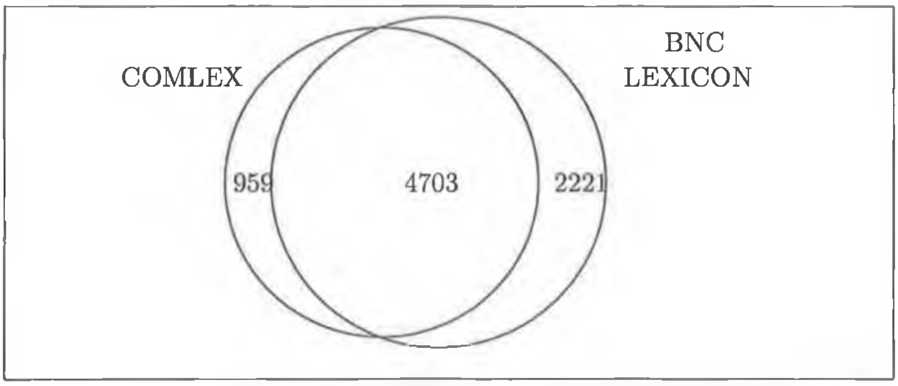


Figure 5.5: Intersection between Active Verb Lemma Types in COMLEX and the Lexicon induced from the BNC (Collins)

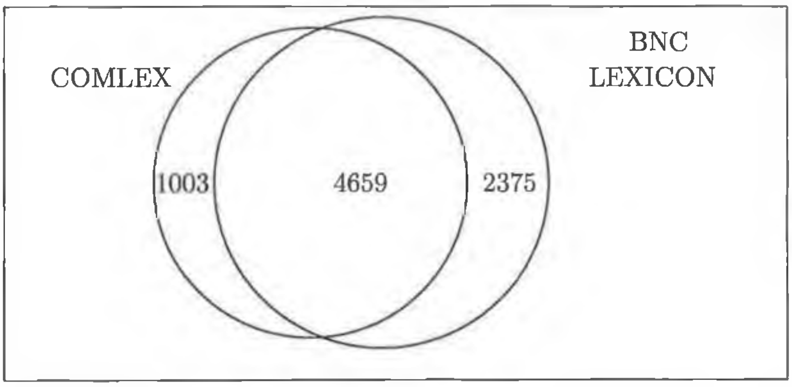


Figure 5.6: Intersection between Active Verb Lemma Types in COMLEX and the Lexicon induced from the BNC (Charniak)

there are over 2000 verb lemmas in the induced lexicon which are not contained in the gold standard. Aside from limited coverage of the gold standard, there are two reasons for the size of this number. First, tagging and lemmatisation errors result in lexical entries for word forms which are not actually verb lemmas. The second reason relates to gold-standard coverage. In particular, the induced lexicons contain a number of verbs with the prefixes `co-`, `re-` and `un-` which are not contained in the gold standard lexicon. For example, the Charniak parser-based lexicon contains entries for 123 lemmas with the prefix `re-` while COMLEX contains entries for only 6 verbs with the same prefix.

Table 5.14 contains the results of evaluating the two induced subcategorisation lexi-

	Collins		Charniak	
# Verbs	4703		4659	
	Induced	Baseline	Induced	Baseline
Precision	62.89	63.27	61.17	63.48
Recall	64.46	43.06	61.67	42.99
F-Score	63.66	51.25	61.42	51.27

Table 5.14: Evaluation of Lexicons induced from BNC Written Component against COMLEX

cons against COMLEX. Following the evaluation methodology presented in Section 4.4, I use COMLEX-LFG Mapping 2 and create a baseline by automatically assigning to each verb the two most frequent frames (transitive and intransitive) and evaluate this artificial lexicon against the gold standard. In both cases the baseline is exceeded by more than 10%. The lexicon induced using the Collins parser achieves an f-score of 63.66% for a lexicon of 4703 verb lemmas. This is 2.24% higher than the f-score achieved using the Charniak parser for a smaller induced lexicon (4659 verb lemmas).

In order to examine lexical quality more closely, I evaluated the induced lexicons for a test set of 100 randomly extracted, commonly occurring verbs with corpus frequency between 1000 and 5000 (cf. Table 5.15). The results against COMLEX are presented in Table 5.16. In this case, the lexicon induced using the Charniak parser as c-structure engine outperforms that extracted using Collins (f-score of 72.84% versus 69.86%). In both cases the baseline is exceeded by over 22%.

Error Analysis

In order to identify potential weaknesses of our approach or shortfalls in gold-standard coverage, I manually examine the set of false positives recorded for each extraction model in the evaluation against COMLEX for 100 frequently occurring verbs (Table 5.15). The Collins evaluation results in 96 false positives (semantic forms) and the Charniak in 88. I first examine the false positives that both systems have in common before examining those unique to each parser.

There are 65 false positives common to both evaluations, 31 only seen in the Collins evaluation and 23 only seen in the Charniak evaluation. During the WSJ experiments

abandon	absorb	account	acknowledge	admire
advance	advise	alter	amount	analyze
assert	block	bother	breathe	brush
cease	celebrate	compete	complain	convey
cook	cope	dance	date	defend
deliver	descend	design	dig	direct
doubt	drift	employ	enhance	exploit
expose	fade	favor	fear	fire
flee	flow	forgive	grasp	hesitate
hurt	illustrate	impose	inform	install
interpret	invest	justify	knock	lack
land	last	leap	limit	link
list	matter	measure	oppose	overcome
plead	possess	practise	pursue	qualify
recover	register	relax	remark	resemble
review	ride	rub	search	select
sink	smell	smoke	sort	stage
steal	stretch	submit	suit	suspect
swear	sweep	switch	test	threaten
trace	transform	undertake	vote	wake

Table 5.15: Test Verbs used for BNC Evaluation against OALD and COMLEX

	Collins		Charniak	
	Induced	Baseline	Induced	Baseline
Precision	75.26	76.00	77.72	76.00
Recall	65.18	33.93	68.53	33.93
F-Score	69.86	46.91	72.84	46.91

Table 5.16: Evaluation of Lexicons induced from BNC Written Component against COMLEX (100 Verbs)

(cf. Section 5.3), extracting obliques was identified as an area where the parser-based approach tended to over-generate. This is reflected in the common set of false positives. Out of 65, 29 are semantic forms whose frames contain obliques. Of these I believe one is correct (`remark([comp,obl])`). Although a higher threshold is used for frames containing obliques, prepositional phrase attachment and argument/adjunct distinctions need to be re-visited in order to improve this aspect of our system. Of the remaining 36 common false positives, 18 are intransitive frames. With only 3 exceptions (the intransitive use of the verbs `dance`, `fade` and `forgive`), these are found to be incorrectly assigned an intransitive frame on manual examination. This is potentially due to the passive use of certain verbs not always being identified and marked correctly by our system and is an area which I hope to improve. The remaining 18 common false positives are found to be correctly rejected by the gold standard.

Looking at the set of 23 false positives unique to the Charniak extraction, a similar pattern emerges to that displayed by the common set. 8 are semantic forms containing obliques and 10 are intransitive semantic forms. Of these, only the following 2 are considered correct after manual examination: `deliver([])` and `swear([comp,obl])`. Interestingly, when I examine the 31 false positives unique to the Collins extraction, I see a different pattern. There are no intransitive semantic forms and 13 semantic forms containing obliques of which two are judged correct (`complain([comp,obl])` and `sort([obj,obl])`). Of the remaining 18 false positives all bar one contain either a COMP or an XCOMP, indicating that this is an area where the Collins-based extraction in particular over-generates. The remaining frame is `forgive([obj,obj2])` which I judge to be correct.

5.5.3 OALD Evaluation

For the OALD evaluation, I restrict the induced lexicons as described in Section 5.5.2 above. I use the OALD mapping and diathesis alternation rules described in Section 4.5. Figures 5.7 (Collins and OALD) and 5.8 (Charniak and OALD) depict the overlap between the induced resources and OALD. The larger evaluation is for the Collins lexicon (4611 lemmas versus 4387 lemmas for Charniak). Against the OALD the Collins parser again outperforms Charniak, achieving an f-score of 61.68%. The lexicon induced using

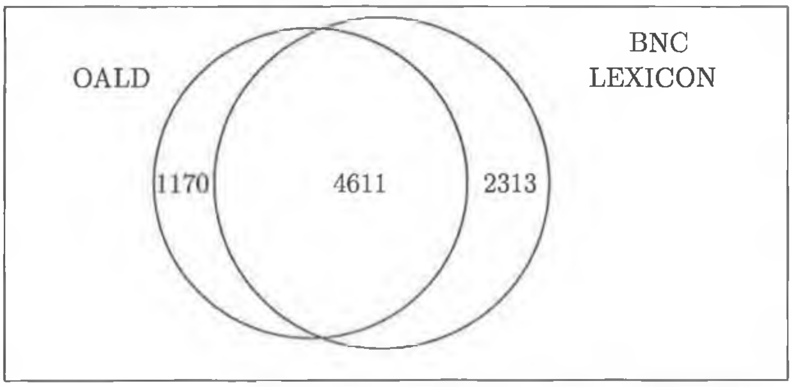


Figure 5.7: Intersection between Active Verb Lemma Types in the OALD and the Lexicon induced from the BNC (Collins)

	Collins		Charniak	
# Verbs	4611		4387	
	Induced	Baseline	Induced	Baseline
Precision	62.39	62.46	59.61	62.81
Recall	60.98	53.13	59.70	52.43
F-Score	61.68	57.42	59.66	57.15

Table 5.17: Evaluation of Lexicons induced from BNC Written Component against the OALD

Charniak is smaller (4387 verb lemmas) with an f-score of 59.66%. In both cases, the baseline is exceeded by more than 2.5 percentage points.

The results achieved against the OALD for the 100 test verbs (cf. Table 5.15) are presented in Table 5.18. In this case, as for the larger evaluations (Tables 5.14 and 5.17), the lexicon induced used Collins Model 2 outperforms that induced using Charniak (f-score of 70.92% versus 68.83%). Both scores exceed the baseline (by 17.92% and 15.83%).

Error Analysis

As for the COMLEX evaluation, I manually examine the false positives recorded for the OALD evaluation using 100 test verbs (Table 5.15) for both Collins and Charniak. Using Collins in the extraction procedure results in 83 false positives while using Charniak results in 93. I first examine the common set of false positives (72) before looking at those unique

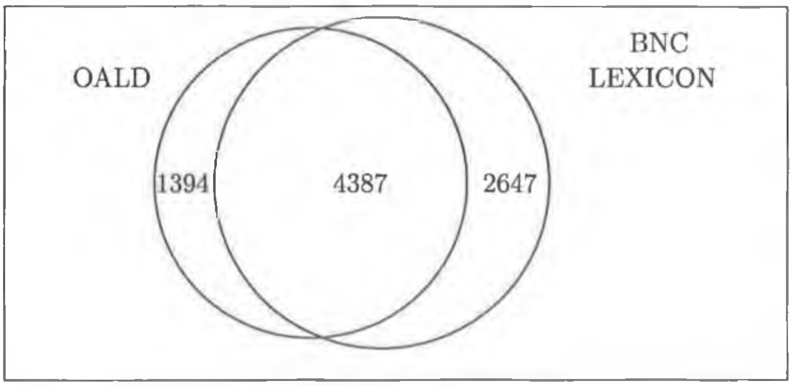


Figure 5.8: Intersection between Active Verb Lemma Types in the OALD and the Lexicon induced from the BNC (Charniak)

	Collins		Charniak	
	Induced	Baseline	Induced	Baseline
Precision	73.31	70.50	70.57	70.50
Recall	68.67	42.47	67.17	42.47
F-Score	70.92	53.00	68.83	53.00

Table 5.18: Evaluation of Lexicons induced from BNC Written Component against the OALD (100 Verbs)

to each parser-based extraction evaluation.

As for the COMLEX error analysis, I find that a lot of false positives contain either intransitive frames (21) or frames containing obliques (23). Of the 21 intransitive false positives manual examination shows that 5 may actually be used intransitively: `celebrate`, `deliver`, `plead`, `submit` and `vote`. The OALD is more conservative than COMLEX in its assignment of frames containing PPs. Of the 23 false positives containing obliques, 8 are found to be correct. In contrast to COMLEX, the OALD under-generates particularly in relation to directional prepositions demonstrated by its rejection of the following frames as incomplete: `drift([obl:pval])`, `flee([obl:pval])` and `flow([obl:pval])`. There is just one example where the alternation rules obviously failed: `forgive([obj,obl:for])`.[12] There are 12 false positives containing

[12] This frame is correct in that `forgive` can occur with an object and an oblique with `for`. However, this is not the benefactive use of the preposition so the frame should have been converted to `forgive([obj,obl:pval])`.

Range	0-15	0-20	0-25	0-30	0-35	0-40	0-50	0-60	0-80
# Verbs	2481	2543	2586	2625	2616	2634	2627	2607	2583
Precision	66.55	65.47	64.23	63.43	62.79	62.43	62.11	61.52	62.15
Recall	63.16	64.63	64.52	64.80	65.06	64.88	64.80	64.59	65.14
F-Score	64.81	65.05	64.38	64.11	63.90	63.63	63.43	63.02	63.61

Table 5.19: # Verbs, Precision, Recall and F-Score against Sentence Length Range (Experiment 1)

Range	5-15	5-20	5-25	5-30	5-35	5-40	5-50	5-60	5-80
# Verbs	2410	2480	2529	2548	2544	2551	2521	2530	2524
Precision	66.63	65.04	63.37	62.77	62.68	62.28	62.42	62.18	61.76
Recall	63.15	64.14	64.13	64.23	64.54	64.54	64.48	64.58	64.20
F-Score	64.84	64.59	63.75	63.49	63.60	63.39	63.44	63.36	62.96

Table 5.20: # Verbs, Precision, Recall and F-Score against Sentence Length Range (Experiment 2)

COMPs which were correctly rejected by the gold standard. However, manual examination showed the following 5 semantic forms containing XCOMPs to be incorrectly rejected by the gold standard: `bother([xcomp(ing)])`, `employ([obj,xcomp(toinf)])`, `justify([xcomp(ing)])`, `threaten([xcomp(toinf)])` and `vote([xcomp(toinf)])`.

Of the 21 false positives which uniquely occur in the Charniak lexicon, 8 are intransitives and 9 contain obliques. Of the 21 frames, manual examination uncovered only one which had been incorrectly rejected, namely the intransitive use of the verb `transform`. There is also another example of an alternation rule failure: `admire([obj,obl:for])`. The COMLEX error analysis revealed that the majority of the false positives unique to the Collins lexicon contained COMPs or XCOMPs. This tendency of the Collins parser-based extraction system is evident in the OALD error analysis also. Out of 11 false positives, 6 contain COMPs or XCOMPs, 3 are intransitive and 3 contain obliques. There is only one which I believe has been incorrectly rejected: `select([obj,xcomp(toinf)])`.

5.6 Effect of Sentence Length on Lexicon Accuracy and Coverage

Typically, parser-based subcategorisation frame extraction is performed on quite re-

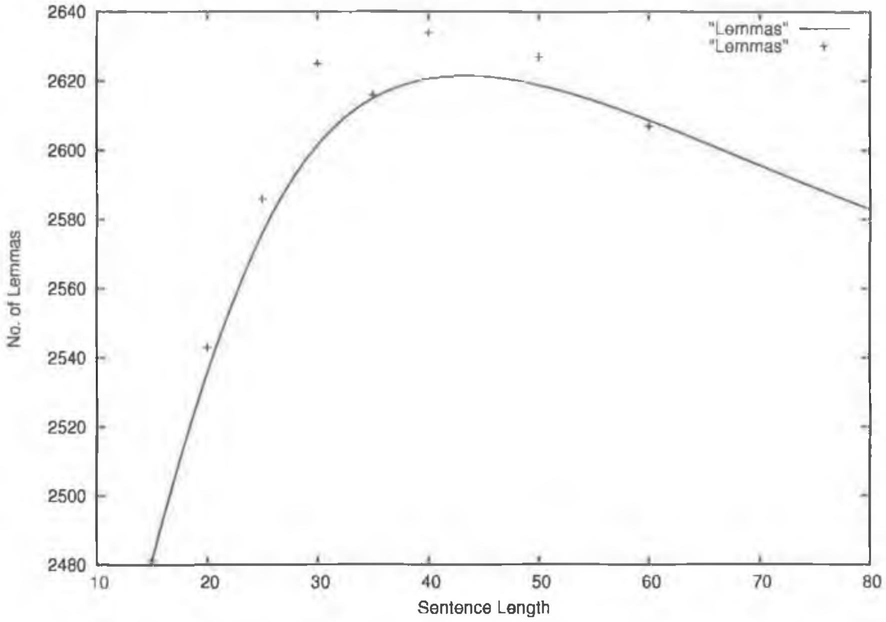


Figure 5.9: Relationship of Extracted Lemmas to Sentence Length for Experiment 1

stricted sentence lengths. Schulte im Walde (2002a), for example, uses sentences of length 5 and 10 for training and length 5 and 13 for extraction. The reason for this restriction is twofold:

1. Full CFG parsing is expensive, typically cubic in sentence length for memorisation-(chart)-based approaches.

2. Shorter strings should be easier to parse with better results and hence produce more reliable (parse-annotated) input for subcategorisation frame extraction.

However, restricting input to shorter strings could potentially carry the risk of (a) missing verb lemmas, (b) missing frame types for verb lemmas and (c) biasing lemma-frame probabilities. To the best of our knowledge, the assumptions (2) and (a) (and for that matter (b) and (c)) above have never been tested empirically in a task-based evaluation.

In order to test assumptions (2), (a) and (b) in a task-based evaluation, namely subcategorisation frame extraction, I carried out two experiments. To achieve meaningful results, corpus size (number of words) was held constant in each of the experiments. In Experiment 1, I investigate the effect of sentence lengths 0-15 up to 80 increasing maxi-

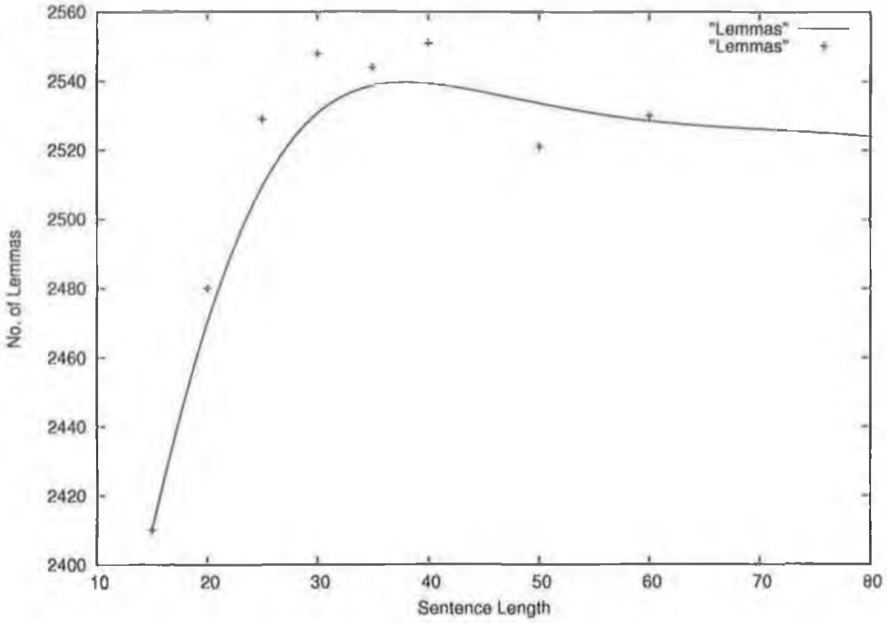


Figure 5.10: Relationship of Extracted Lemmas to Sentence Length for Experiment 2

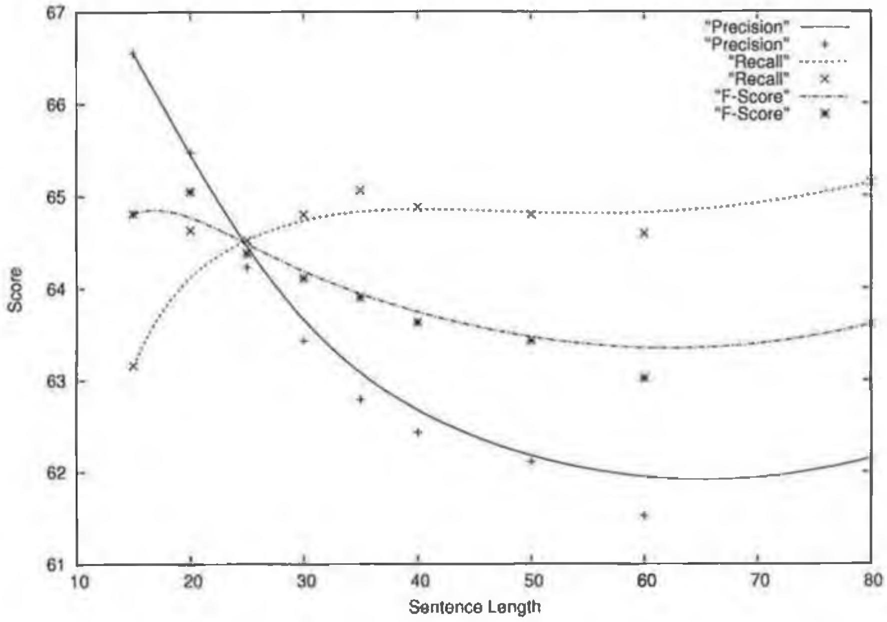


Figure 5.11: Relationship of Precision, Recall and F-Score to Sentence Length for Experiment 1

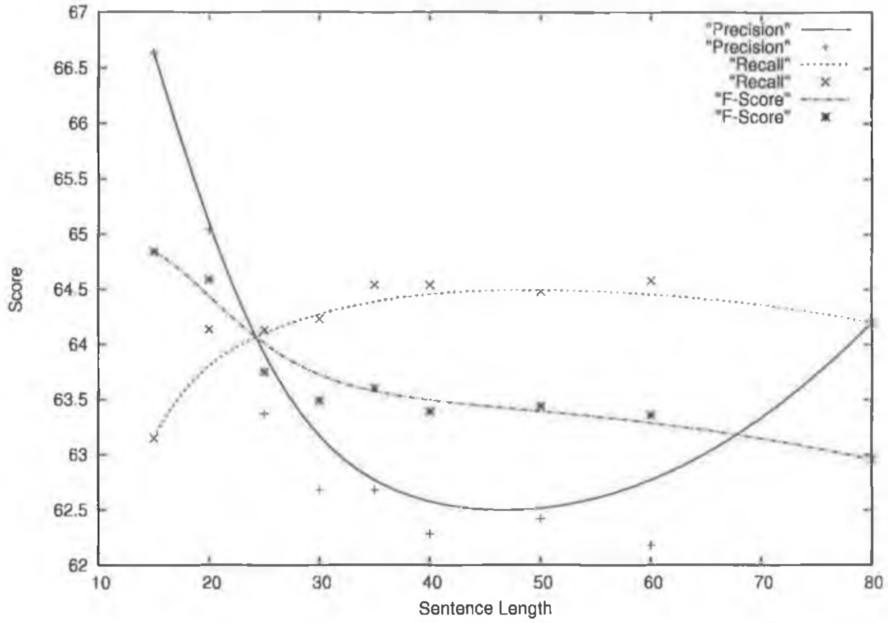


Figure 5.12: Relationship of Precision, Recall and F-Score to Sentence Length for Experiment 2

mum length in steps of five. Maximum corpus size is determined by the total number of sentences of length 0-15 in the BNC: 13 million words. For each five-word sentence length increase, I randomly extract sentences between 0 and the upper bound from the BNC until the desired corpus size of 13 million words is reached. If sentence length restricts the number of lemmas, then we expect the size of the lexicon in terms of headwords to grow with increased sentence length. Subcategorisation frame recall is expected to grow with sentence length while precision is expected to suffer. Experiment 2 is similar to Experiment 1 but in this case the lower sentence length bound is 5 rather than zero as sentences less than length 5 are potentially incomplete or ungrammatical.

Table 5.19 summarises the results of Experiment 1 and Table 5.20 summarises those of Experiment 2. The results are represented graphically in Figures 5.9, 5.10, 5.11 and 5.12. Figures 5.9 and 5.10 illustrate the number of lemmas in the extracted lexicon in relation to their sentence length range. The data points are included and a normalised curve (using Bezier Smoothing) represents the general trend. In both cases, the number of lemmas peaks at about the 40-word mark before decreasing again with the extension of the sentence length range. Figures 5.11 and 5.12 plot precision, recall and f-score against

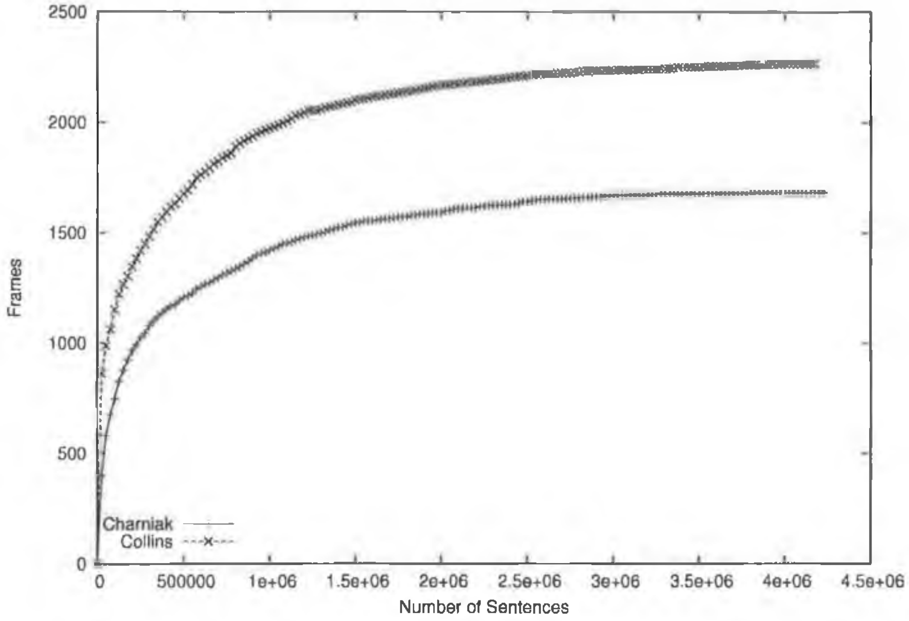


Figure 5.13: Accession Rate of Frames (with Prepositions and Particles, Frequency > 1)

sentence length. In both cases precision initially drops with the increase of sentence length while recall rises. In the case of Experiment 1 (Figure 5.11), both precision and recall level out after initial fluctuation. Recall behaves similarly in the case of Experiment 2 (Figure 5.12), while precision begins to rise again with the increase in sentence length after the 50-word mark.

These results are interesting and the general trend supports assumptions 2 and (a) above. However, I feel that further careful experimentation is required in order to draw more significant conclusions from this preliminary investigation.

5.7 Lexical Accession Rates

I carried out an extensive quantitative evaluation of the induced resources in order to establish coverage on unseen text and demonstrate the lexical accession rate of our system. Figure 5.13 shows the growth rate of frame types parameterised for prepositions and particles, while Figure 5.14 is for frames without this information. In both cases, I filter out noisy frames by only including those which are seen more than once. There

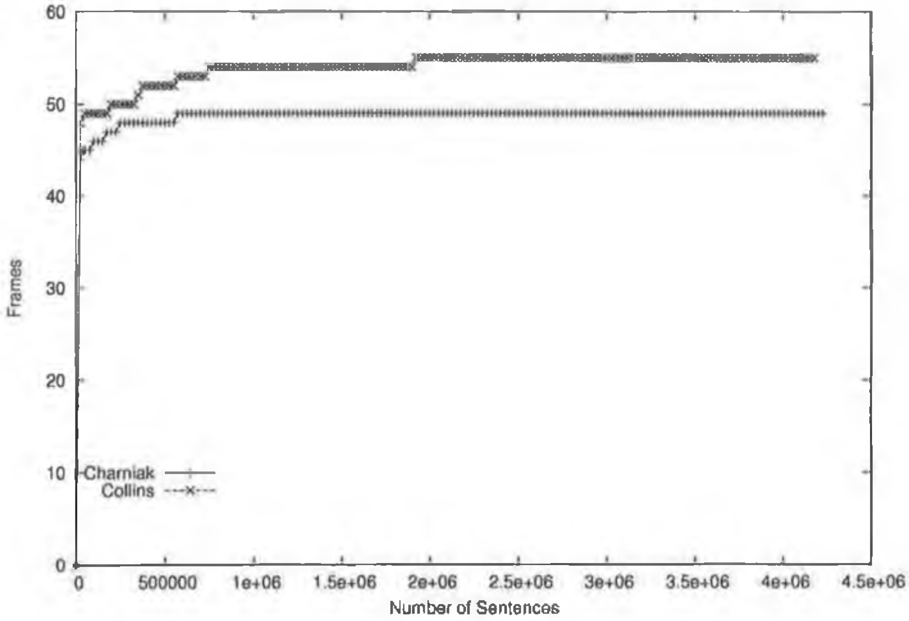


Figure 5.14: Accession Rate of Frames (without Prepositions and Particles, Frequency > 1)

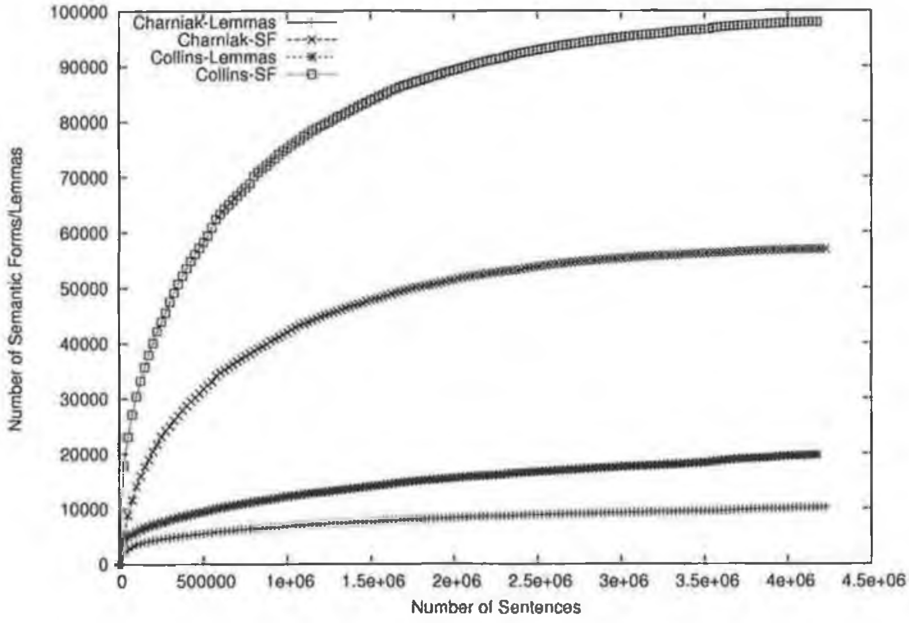


Figure 5.15: Accession Rate of Semantic Forms and Lemmas (Frequency > 1)

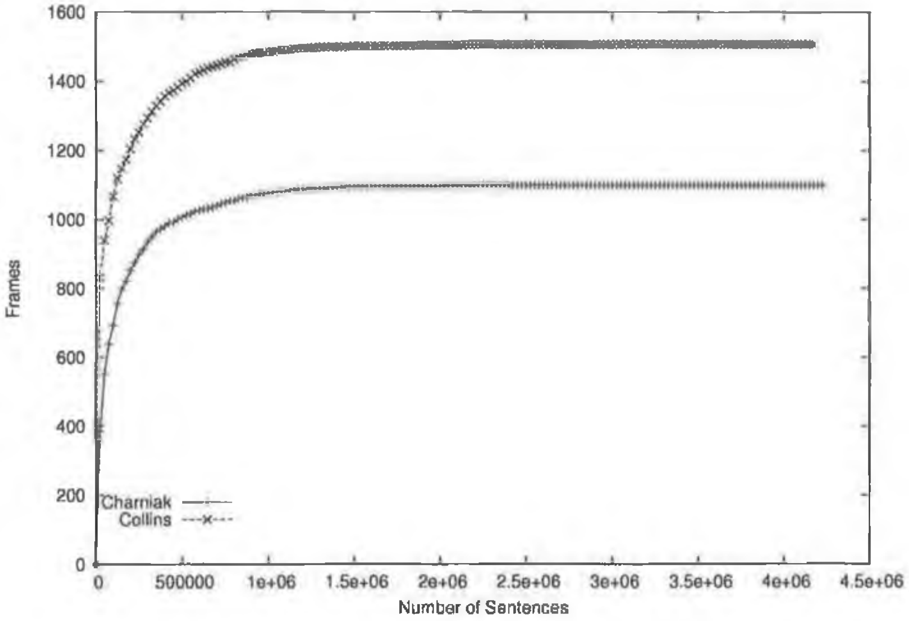


Figure 5.16: Accession Rate of Frames (with Prepositions and Particles, Frequency > 5)

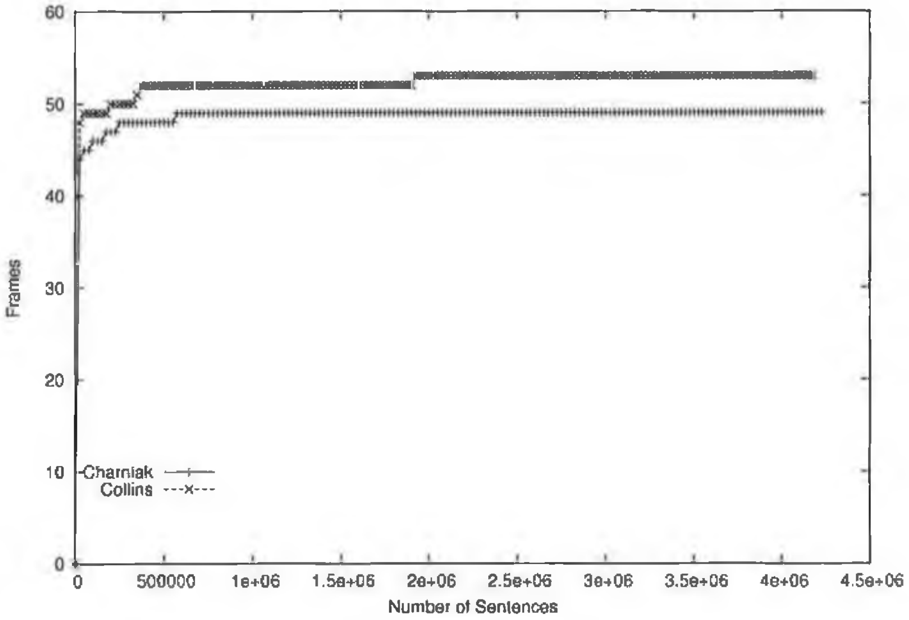


Figure 5.17: Accession Rate of Frames (without Prepositions and Particles, Frequency > 5)

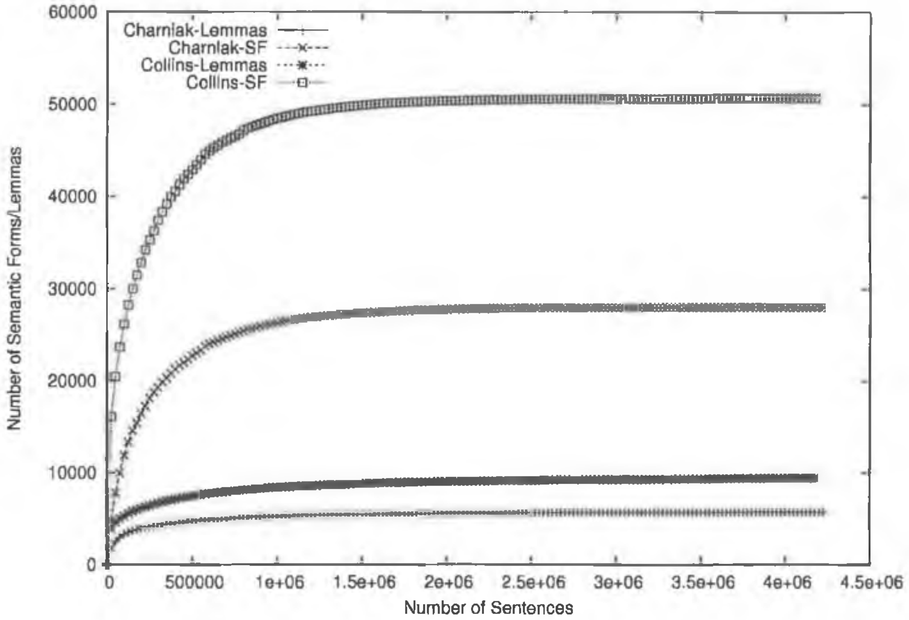


Figure 5.18: Accession Rate of Semantic Forms and Lemmas (Frequency > 5)

is an obvious levelling out in each case, more pronounced in Figure 5.14. Figure 5.15 shows the accession rates for lemmas and semantic forms (lemma/frame combinations). The same filtering is applied. Here too the accession rate slows down although less so for semantic forms. Raising the filter to an absolute threshold of 5 (as used in the qualitative evaluation, Section 5.5.1) accentuates this levelling out of the curves or the ever decreasing rate of acquisition of new material as corpus size grows (cf. Figures 5.16, 5.17 and 5.18). The trends demonstrated by frame induction and lemma induction imply that the steeper semantic form curve is due predominantly to the discovery of combinations of previously seen frame types with previously seen verbs.

It is interesting to compare the empirically established results with the combinatorial possibilities for generating frame types. Given a set of grammatical functions $G = \{$SUBJ, OBJ, OBJ2, OBL, OBL2, COMP, XCOMP, PART$\}$ (as used in the experiments in this thesis and not parameterised for particular prepositions and particles) and a maximum frame size (length) of 3, the set of possible frames is given by:

$$\sum_{n=1}^{3} \frac{|G|!}{n!(|G|-n)!} = \frac{8}{1} + \frac{8 \cdot 7}{2} + \frac{8 \cdot 7 \cdot 6}{6} = 92$$

Out of these, the experiments in Table 5.11 show that the system automatically induces 55 frame types using Collins and 50 using Charniak.

5.8 Summary

In this chapter I show how the lexical extraction methodology presented in Chapter 3 can be incorporated into a system for the extraction of lexical resources from raw text. The system architecture combines the f-structure-based semantic form extraction methodology with technology described in (Cahill et al., 2004b) for the parsing of text into f-structures and introduces a new way of acquiring 'safe' semantic forms from proto f-structures for LDD resolution. First the text is tagged and parsed. The modularity of the architecture facilitates the combination of different parsers and taggers. The next step in the extraction process is the automatic f-structure annotation of the parse trees, resulting in a set of 'proto' f-structures (with LDDs unresolved). For each 'proto' f-structure, I extract 'safe' semantic forms from sub-f-structures which do not contain an unresolved LDD. This set of semantic forms is then added to the set of semantic forms extracted from Sections 02-21 of the Penn-II Treebank (the training set of the statistical parsers) and the conditional probabilities are recalculated. The LDDs of the 'proto' f-structures are then resolved using this extended set of semantic forms to produce proper f-structures. From the proper f-structures I extract the final semantic form lexicon, which now includes semantic forms reflecting the resolved LDDs in the source f-structures. In order to examine the reliability of the system architecture, I extract and evaluate lexicons from the strings of Sections 00, 01, 22, 23 and 24 using Charniak's (2000) maximum entropy inspired parser, Collins' (1999) Models 2 and 3, and Bikel's (2002) history-based parser (both the original version and a version retrained to retain Penn-II functional tags) in combination with MXPOST (Ratnaparkhi, 1996) and TreeTagger (Schmid, 1994). I compare the parser-induced subcategorisation lexicons to a lexicon induced from the original gold-standard treebank trees. Surprisingly and encouragingly, the lexicon extracted from the treebank trees does not perform statistically significantly better than any of the lexicons induced from raw text in an evaluation against COMLEX.

Extending the basic extraction methodology to process raw text provides the opportunity to compare the extraction approach to a state-of-the-art system, namely that of Korhonen (2002). I describe the mapping from our frames to the (Korhonen, 2002) gold standard and extract frames from the 65,000-word (Korhonen, 2002) test corpus using the Penn-II-based LFG parsing architecture with the TreeTagger, Bikel's parsing engine (retrained to retain Penn-II functional tags) and the f-structure annotation algorithm. I include passive and diathesis alternation rules to improve the quality and coverage of the induced lexicon. The best f-score for the 29 test verbs is 76.16% against 71.46% when I replicate Korhonen's best results for the same verbs. This is a statistically significant improvement.

In order to assess the robustness and scalability of the parser-based approach, I use a 90 million-word corpus of BNC text to extract large-scale lexicons using Charniak's and Collins' Model 2 parsers with the TreeTagger, and evaluate the lexicons against the OALD and COMLEX. The evaluations range in size from 4387 to 4703 test verbs, the most extensive to my knowledge that have been carried out for English. In addition, I carry out smaller evaluations for a set of 100 frequently occurring test verbs with both lexicons achieving over 70% against COMLEX. In all cases, the induced lexicons exceed the baseline in evaluation against the gold standards.

Using a large corpus such as the BNC as the data source for our lexical acquisition methodology permitted me to examine the effect of sentence length on induced lexicon coverage and quality. While the trends manifest in the results confirmed our expectations, i.e. that the number of lemmas in the lexicon increases with corpus size while the accuracy of the lexical entries drops, I believe that further investigation is required to draw significant conclusions about this matter.

Finally, I examine the rate of accession of lexical information over this large corpus. Judging by the slowing rate at which new frame types and lemma types are induced, I conjecture that the main reason for the growth in semantic forms is the combination of existing lemmas with different but previously seen frames.

Chapter 6

Multilingual Treebank-Based Grammar and Lexicon Extraction

6.1 Introduction

Manual construction of rich grammatical and lexical resources, particularly multilingual resources, is time-consuming, expensive and requires considerable linguistic and computational expertise. In our work, we have developed an approach which exploits information encoded in treebank trees to automatically annotate each node in each tree with f-structure equations representing abstract predicate-argument structure relations. From the annotated treebank, we automatically extract large-scale unification grammar resources, namely probabilistic LFG approximations, and subcategorisation information, for parsing new text into f-structures. A growing number of treebanks for languages other than English (including Japanese, Chinese, German, French, Czech and Spanish) are becoming available. (Cahill et al., 2003), (Cahill et al., 2005b) and (Burke et al., 2004c) show how the lexical and grammatical extraction approaches described in (Cahill et al., 2004b) and (O'Donovan et al., 2004) for English can be successfully migrated to typologically different languages (German and Chinese) and different treebank encodings (TIGER and Penn CTB). In this chapter, I will describe the porting of the methodology to Spanish and the Cast3LB Treebank (Civit, 2003). I present an f-structure annotation algorithm for Cast3LB and describe how LFG resources for Spanish can be induced from the f-structure-annotated treebank. I extract PCFG-based LFG approximations and report on a number of parsing experiments. I evaluate both the quality of the automatic f-structure annotation of the Cast3LB treebank, and the parser output. I describe how lexical resources can be extracted from the f-structure-annotated treebank and present sample lexical entries. Finally I summarise the work carried out for German and Chinese with particular emphasis on the induced lexical resources. The work reported on Spanish here was published as (O'Donovan et al., 2005b).

6.2 From Cast3LB to a Spanish LFG

6.2.1 Cast3LB Treebank

The Cast3LB treebank (Civit, 2003) consists of 125,000 words (approximately 3,500 trees) taken from a wide variety of Spanish texts (journalistic, literary, scientific) from both Spain and South America. Despite the free word order of Spanish, constituency rather then dependency annotation is used in the Cast3LB treebank. Unlike the Penn-II Treebank which loosely complies with X-bar theory, the phrase-structure trees of the Spanish Treebank are essentially theory-neutral. Only lexically realised constituents are annotated with the exception of elided subjects in pro-drop constructions. There are, therefore, no empty nodes and traces unlike in the Penn-II Treebank. Another policy of the Cast3LB creators was not to alter the surface word order of the constituents. Due to the free word order of Spanish, a verb phrase containing the verb and its arguments (other than subject) cannot always be established. As a result the main constituents of the sentence are daughters of the root node. The free word order of Spanish also means that phrase-structural position is not an indication of grammatical function, a feature of English which was heavily exploited in the automatic annotation of the Penn-II Treebank. Instead I take advantage of the rich Cast3LB functional annotation of verbal dependants and the fine-grained non-terminals to annotate the treebank with f-structure equations.

Figure 6.1 shows an example tree from the Cast3LB Treebank. The verbal elements of the sentence are realised by the gv (grupo verbal) subtree. The sn (sintagma nominal) subject of the sentence is marked as such using the functional tag SUJ. Any other verbal

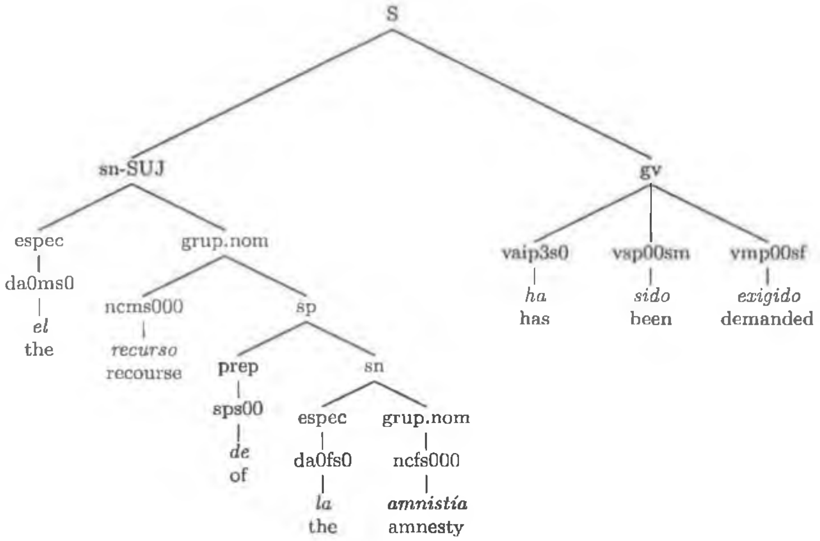


Figure 6.1: Example Tree from the Cast3LB Treebank

complements and adjuncts are marked in a similar way in the treebank. The full list of functional labels is provided in Table 6.1. Constituents which are not verbal complements do not receive functional annotations. The full list of phrasal category labels (i.e. excluding preterminals) is presented in Table 6.2. In addition to these, any of the clausal nodes may be annotated with an asterisk to indicate verbal ellipsis in coordinated structures. The tree in Figure 6.2 where the verb *es* is omitted from the second conjunct demonstrates this phenomenon (cf. S*). The preterminal tags in Cast3LB are fine-grained (see Figures 6.1 and 6.2) because they encode morphological as well as part of speech information. For example the tag ncms000 indicates that *recurso* is a common noun which is masculine and singular.

6.2.2 Automatic Annotation of Cast3LB Trees

The annotation algorithm for Spanish is constructed following the same methodology used for English, German and Chinese. I begin by automatically extracting all the rules and their associated frequencies from the treebank. I extract 7972 rules when preterminals containing morphological information are conflated to basic POS tags.[1] I then select the most frequent rule types for each left hand side (lhs) category which together give

[1] For example the preterminals ncms000 and ncfs000 are conflated to the generic POS tag n.

SUJ	Subject
CD	Direct Complement
CI	Indirect Complement
ATR	Attributive
CPRED	Predicative Complement
CAG	Agentive Complement
CREG	Prepositional Phrase Complement
CC	Adjunct
ET	Textual Element
MOD	Modal Adverb
NEG	Negative
PASS	Passive
IMPERS	Impersonal
VOC	Vocative

Table 6.1: Functional Annotations used in the Cast3LB Treebank

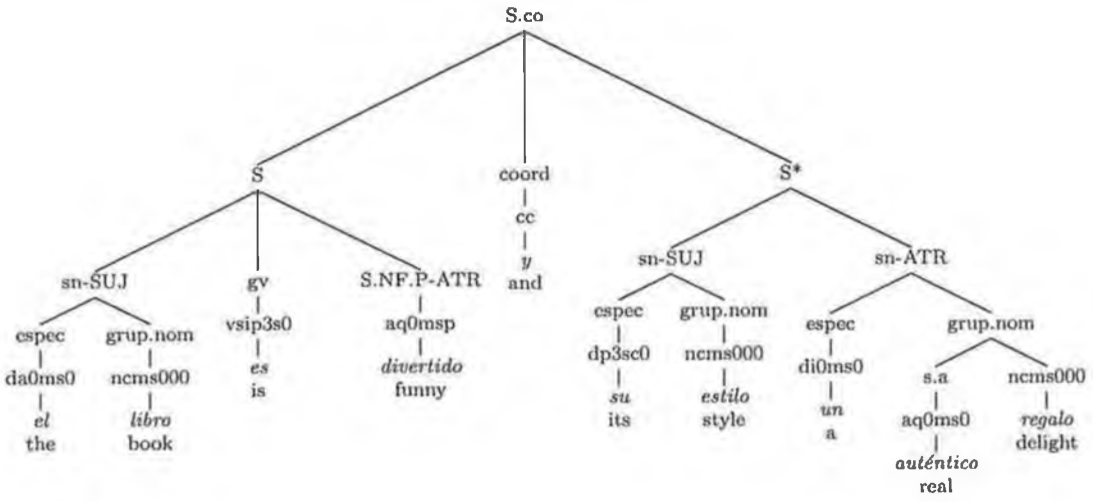


Figure 6.2: Cast3LB Annotation of Verbal Ellipsis in Coordinated Constructions

S.F.C	Subordinated Finite Complement
S.F.R	Subordinated Finite Adjectival
S.F.A	Subordinated Finite Adverbial
S.F.A.Cond	Subordinated Conditional Finite Adverbial
S.F.A.Conc	Subordinated Concessive Finite Adverbial
S.F.A.Cons	Subordinated Consecutive Finite Adverbial
S.F.A.Comp	Subordinated Comparative Finite Adverbial
S.NF.C	Subordinated Non-Finite Complement
S.NF.A	Subordinated Non-Finite Adverbial
S.NF.P	Subordinated Non-Finite Adjectival
S.NF.R	Subordinated Non-Finite Relative
INC	Parenthetical
sn(.e)	Noun Phrase (elided)
sa	Adjectival Phrase
sadv	Adverbial Phrase
sp	Prepositional Phrase
gv	Verbal Group
infinitiu	Infinitival
gerundi	Gerund
grup.nom	Nominal group
prep	Preposition
interjeccio	Interjection
neg	Negation (no)
relatiu	Relative Pronoun
numero	Number
morfema.verbal	Pronoun *se* in passive and impersonal constructions
morf.pron	Reflexive Pronoun
espec	Specifier

Table 6.2: Phrasal Categories from the Cast3LB Treebank

85% coverage of all rule tokens expanding that category. This results in a reduced set of 3638 rules. The right hand sides (rhs) of these 3638 rules are then automatically assigned default annotations, e.g. any node with a SUJ functional annotation is assigned the functional equation ↑SUBJ=↓. The rules are also head-lexicalised following the head finding rules I developed for Spanish (Appendix A). The reason for the relatively large number of CFG rules is the fine-grained tags for sentential nodes which are used in the treebank (Figure 6.2). Of the 3638 rule types, 3533 have a sentential node on the lhs. As many of the daughters of sentential nodes are tagged with Cast3LB functional tags, the rhs of 2870 of the 3638 rules are unsurprisingly completely annotated after automatic head lexicalisation and default annotation. Out of a total of 15039 rhs nodes, 14091 (93.70%) are assigned an annotation automatically. Next the remaining partially annotated rules (768 in total) are manually examined and used to construct annotation matrices which generalise to unseen rules. The annotation matrices encode information about the left and right context of a rule's head. For example, an **espec** node to the left of the head of an **sn**'s head is a specifier while an **sp** node to the right of a **grup.nom**'s head is an adjunct. Lexical information is provided by macros which are written for the POS tags.

I implemented the f-structure annotation algorithm in Java following a similar architecture to that used for English, German and Chinese. The automatic annotation of the entire treebank is essentially a four step process illustrated in Figure 6.3. First, the annotation algorithm attempts to assign an f-structure equation to each node in the tree based on the Cast3LB functional labels. I have compiled an f-structure equation look-up table which assigns default f-structure equations triggered by each Cast3LB functional label. For example, the default entry for the SUJ label is ↑SUBJ=↓. Table 6.3 gives the complete set of default annotations. Next, the head of each local subtree of depth one is found following the head lexicalisation rules I have compiled (cf. Appendix A). For example, the **prep** daughter of an **sp** (prepositional phrase) node is its head and is assigned the f-structure equation ↑=↓. In the third step, the annotation algorithm deals specifically with coordination as this phenomenon is not covered by the left-right generalisations for other constructions. Figure 6.4 provides an example of coordination in the Cast3LB Treebank. The **.co** suffix on the **grup.nom** node label indicates that the node is mother of two or

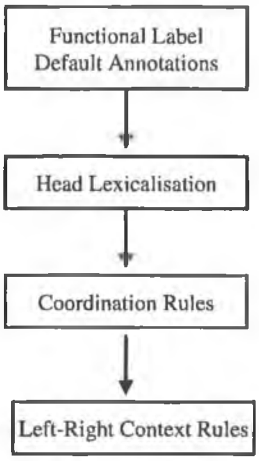


Figure 6.3: Architecture of Spanish Annotation Algorithm

more coordinated `grup.nom` nodes. The coordinating conjunction (`cc`) is annotated as the head of the coordinated noun phrase and the coordinated elements are annotated as elements of the noun phrase's conjunct set. In a final step, the annotation algorithm moves top-down left-to-right through each tree and any unannotated nodes in each local subtree of depth one are assigned f-structure equations using the left-right context principles constructed by examining the subset of most frequent treebank rules mentioned above. For example, an `sn` node to the right of the head of an `sp` is annotated as the object of the prepositional phrase ($\uparrow$OBJ=$\downarrow$). The f-structure equations are then automatically collected and passed to a constraint solver which produces an f-structure. The annotated tree and resulting f-structure for the tree in Figure 6.1 is shown in Figure 6.5. The tense, number and gender information is derived from the lexical macros. At present I produce "proto" f-structures (with unresolved long distance dependencies) rather than "proper" f-structures as the Cast3LB does not contain trace information.

6.2.3 Evaluation of the Annotation Algorithm

I first evaluated the coverage of the annotation algorithm on the entire Cast3LB Treebank. The results are presented in Table 6.4. 96.04% of the sentences receive one covering and connected f-structure. Ideally, I wish to generate just one f-structure per sentence. A number of sentences (102) receive more than one f-structure fragment. This is due to cases

SUJ	↑SUBJ=↓
CD	↑OBJ=↓
CI	↑OBJ_THETA=↓
ATR	↑XCOMP=↓
CPRED	↑XCOMP=↓
CAG	↑OBLAG=↓
CREG	↑OBL=↓
CC	↓∈(↑ADJ)
ET	↓∈(↑ADJ)
MOD	↓∈(↑ADJ)
NEG	↓∈(↑ADJ)
PASS	↑PASSIVE=+
IMPERS	↑IMPERSONAL=+
VOC	↓∈(↑ADJ)

Table 6.3: Functional Tag-Triggered Default Annotations used in the Cast3LB Treebank

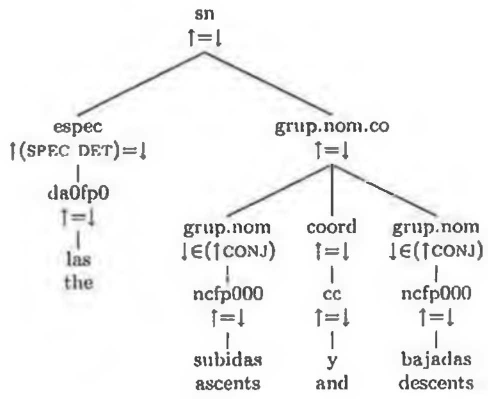
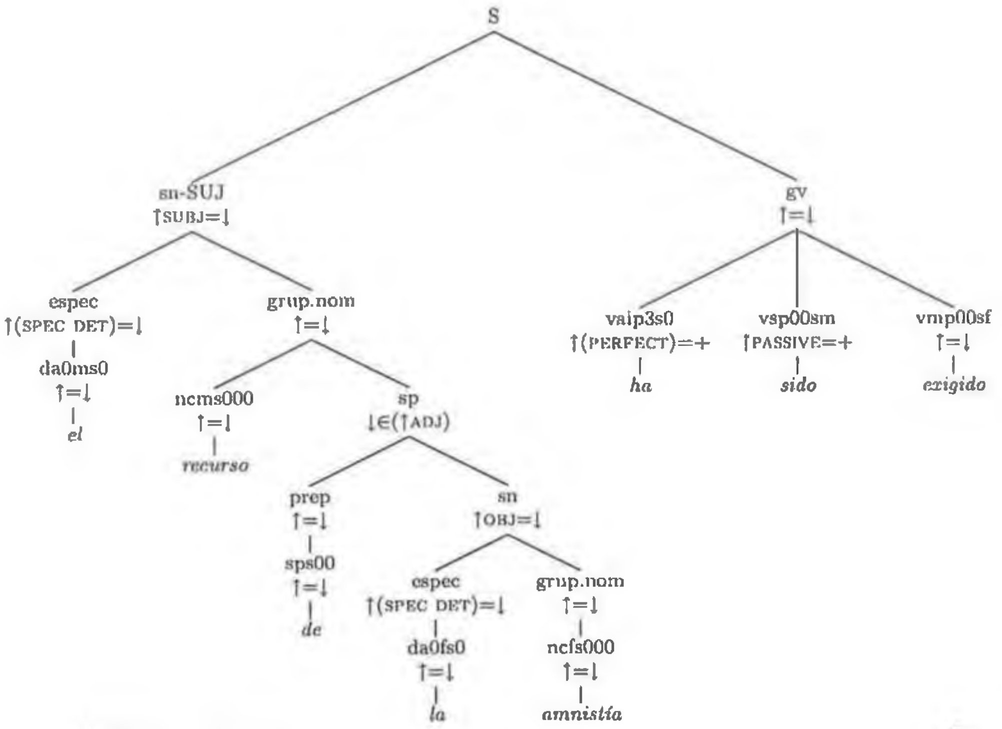


Figure 6.4: Coordination Example from Cast3LB with Automatically Generated F-structure Equations



$$\left[\begin{array}{ll}
\text{PRED} & \text{`EXIGIR'}\\
\text{PERFECT} & +\\
\text{PASSIVE} & +\\
\text{TENSE} & \text{PRESENT}\\
\text{SUBJ} & \left[\begin{array}{ll}
\text{SPEC} & \left[\begin{array}{ll}\text{DET} & \left[\begin{array}{ll}\text{PRED} & \text{`EL'}\\ \text{NUM} & \text{SING}\\ \text{GEN} & \text{MASC}\end{array}\right]\end{array}\right]\\
\text{PRED} & \text{`RECURSO'}\\
\text{NUM} & \text{SING}\\
\text{GEN} & \text{MASC}\\
\text{ADJ} & \left\{\left[\begin{array}{ll}
\text{PRED} & \text{`DE'}\\
\text{OBJ} & \left[\begin{array}{ll}
\text{SPEC} & \left[\begin{array}{ll}\text{DET} & \left[\begin{array}{ll}\text{PRED} & \text{`EL'}\\ \text{NUM} & \text{SING}\\ \text{GEN} & \text{FEM}\end{array}\right]\end{array}\right]\\
\text{PRED} & \text{`AMNISTÍA'}\\
\text{NUM} & \text{SING}\\
\text{GEN} & \text{FEM}
\end{array}\right]
\end{array}\right]\right\}
\end{array}\right]
\end{array}\right]$$

Figure 6.5: Automatically Annotated Tree and F-structure for the Example in Figure 6.1

F-Structures	Trees	% Trees
0	36	1.03
1	3347	96.04
2	96	2.75
3	5	0.14
4	1	0.03

Table 6.4: Coverage and Fragmentation Results of Spanish F-structure Annotation Algorithm

where the algorithm cannot establish a relationship between all elements in the treebank sentence and leaves nodes unannotated. There are also a small number of sentences (36) which do not receive any f-structure. These are a result of feature clashes in the annotated trees, which are caused by inconsistent annotation. Work is ongoing to increase the number of sentences which receive one complete, covering f-structure.

I also evaluate the quality of the annotation against a manually constructed gold standard of 100 f-structures. For the parsing experiments I set aside approximately 10% of the treebank (336 sentences) for testing purposes. This test set is selected randomly from the various text genres which make up the treebank. From this test set, I extracted 100 sentences (cf. Appendix B) at random, to develop the smaller gold standard. The f-structures from the original Cast3LB trees for these sentences generated by the automatic annotation algorithm were manually corrected and converted into dependency format. I use the triples encoding and evaluation software of Crouch et al. (2002). Table 6.5 shows that currently the automatic annotation algorithm achieves an f-score of 95.92% for all grammatical functions and 95.02% for preds-only. In both cases, precision is about 5% higher than recall, indicating that the algorithm at present tends to be more partial than incorrect. Table 6.6 shows a more detailed analysis of how well the automatic f-structure annotation algorithm performs for each function in the all grammatical functions evaluation. The algorithm performs well on most features, e.g. the OBJ f-score is 94% and that for SUBJ is 92%. At present, the worst score (57%) is for the OBLAG feature (the agent in a passive construction), but there are only four occurrences of this feature in the gold standard. I expect this along with all the other figures to improve as the annotation algorithm is further refined.

	Precision	Recall	F-Score
All GFs	98.40	93.56	95.92
Preds-Only	97.90	92.31	95.02

Table 6.5: Evaluation of the Automatically Produced F-structures against the 100 Gold-Standard F-structures

DEPENDENCY	PRECISION	RECALL	F-SCORE
ADJUNCT	608/618 = 98	608/648 = 94	96
AUX	22/22 = 100	22/25 = 88	94
CASE	12/12 = 100	12/17 = 71	83
COMP	21/22 = 95	21/23 = 91	93
CONJ	185/190 = 97	185/196 = 94	96
DET	326/328 = 99	326/342 = 95	97
FORM	56/57 = 98	56/59 = 95	97
GEN	914/920 = 99	914/954 = 96	98
IMPERSONAL	3/3 = 100	3/3 = 100	100
NUM	1115/1130 = 99	1115/1174 = 95	97
OBJ	429/444 = 97	429/464 = 92	94
OBJ_THETA	17/17 = 100	17/19 = 89	94
OBL	13/14 = 93	13/15 = 87	90
OBLAG	2/3 = 67	2/4 = 50	57
PART	4/4 = 100	4/5 = 80	89
PARTICIPLE	27/27 = 100	27/30 = 90	95
PASSIVE	11/11 = 100	11/12 = 92	96
PERS	189/196 = 96	189/207 = 91	94
REFLEX	17/17 = 100	17/18 = 94	97
RELMOD	34/34 = 100	34/36 = 94	97
SUBJ	255/258 = 99	255/294 = 87	92
SUBORD	50/50 = 100	50/54 = 93	96
SUBORD_FORM	50/50 = 100	50/54 = 93	96
TENSE	183/187 = 98	183/196 = 93	96
XCOMP	62/66 = 94	62/73 = 85	89

Table 6.6: Breakdown of All-Grammatical-Functions Annotation Algorithm Evaluation Results by Dependency

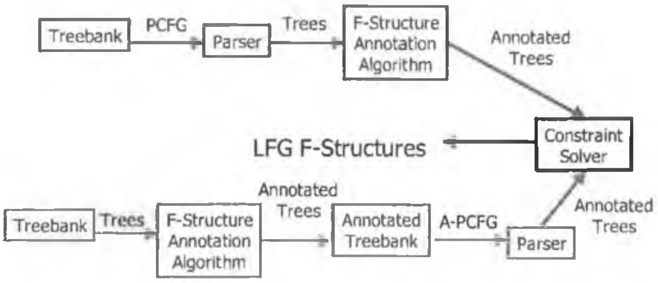


Figure 6.6: Pipeline and Integrated Parsing Architectures

6.3 Parsing Experiments

To parse raw text into f-structures, I use the **pipeline** and **integrated** parsing architectures of Cahill et al. (2004b), illustrated in Figure 6.6. For the pipeline model, I first extract a PCFG from the Cast3LB treebank excluding the 336 test sentences. Cast3LB functional tags are retained in the grammar extraction. I use Helmut Schmid's BitPar parser (Schmid, 2004) to parse new text with the grammar, using Viterbi pruning to obtain the most probable parse. The resulting parse trees are then automatically annotated using the annotation method described in Section 6.2.2. The f-structure equations are collected from the trees and passed to the constraint solver which produces an f-structure for each sentence. For the integrated model, I first automatically annotate the Cast3LB treebank with f-structure equations. I then read off a grammar from the annotated treebank, resulting in an *annotated* PCFG (A-PCFG) for Spanish. I again use BitPar to parse new text with this grammar producing annotated trees. Again the f-structure equations are collected from the parse trees and passed to the constraint solver to produce f-structures. I also transformed each grammar using a parent transformation (Johnson, 1999) to give a P-PCFG and a PA-PCFG.

In addition, I extend Dan Bikel's multilingual, parallel-processing statistical parsing engine (Bikel, 2002) to include a language package for Spanish. Implemented in Java, the parsing engine is a history-based parser emulating Collins' Model 2 (Collins, 1997). The language package is a collection of Java classes that are extensions of several of the abstract classes which provide the description of data and methods specific to a particular language and treebank annotation style. Aside from creating the Spanish classes, I added a data file specifying the head rules specific to the Spanish Cast3LB treebank to be read by the HeadFinder class. With this extension, I trained the parser on the training set of the treebank retaining Cast3LB functional tags and parsed the test set with the grammar. Following the pipeline model, I then automatically annotated the resulting parse trees, collected the f-structure equations and passed them to the constraint solver to produce f-structures.

As previously noted, the Cast3LB preterminals are very fine-grained, encoding extensive morphological detail in addition to POS information. For example, the tag `vaip3s0` denotes a verb (`v`) which is an auxiliary (`a`), used indicatively (`i`) in the present tense (`p`), and is third person (`3`) singular (`s`). In total there are 327 preterminal types in the treebank. This level of fine-grainedness together with the relatively small training set causes a data sparseness issue for parsing new text. With such a large number of POS tags, it is inevitable that certain tags appear in the test set which have not been seen in a similar context in training with adverse effects on coverage.[2] To deal with this issue, initially I masked the morphological detail in the preterminals by conflating them to more generic POS tags.

6.3.1 Initial Results

I then parsed the 336 raw test sentences with the four grammars using BitPar and the retrained and extended Bikel parsing engine. The results are shown in Table 6.7. I evaluated the quality of the trees produced by the parsers using `evalb` and measured how many of the 336 sentences produce one covering and connected f-structure. The PCFG performs best in terms of coverage and fragmentation with over 96% of sentences being assigned one covering and connected f-structure. Coverage drops for the A-PCFG with fragmentation of 93.64%. This trend continues when parent transformations are added (71.21% for PA-PCFG). This may be attributed to data sparseness problems. The PA-PCFG rules are very information-rich and it is possible that constructions encountered in testing will not have been seen during training. As before, I qualitatively evaluated the automatically produced f-structures against the manually constructed gold standard

[2] If BitPar encounters a sentence in the test set containing a previously unseen tag, it will crash at that point.

	PCFG	A-PCFG	P-PCFG	PA-PCFG	Bikel
Parses (out of 336)	334	330	305	264	328
Labelled F-Score	79.01	78.89	78.78	78.44	79.19
Unlabelled F-Score	82.64	82.45	82.61	81.86	82.28
Fragmentation (336 F-Structures)	96.11	93.64	85.90	71.21	88.41
All GFs (100 F-Structures)	59.70	57.99	55.75	46.93	60.13
Preds-Only (100 F-Structures)	69.38	68.01	66.02	55.88	72.11

Table 6.7: Initial Parsing Results for Spanish

using the evaluation software of Crouch et al. (2002). The results of this evaluation reveal a problem with the use of preterminal conflation to avoid data sparseness problems in parsing. Usually an all-grammatical-functions evaluation is less rigid than a preds-only evaluation as the features with atomic values (such as person, number and gender) are typically associated with the correct local `pred` even if the `pred` is attached incorrectly in global f-structure. In the case of these experiments, however, the grammars score very poorly (as low as 46.93% for the PA-PCFG) in the all-grammatical-functions evaluation. By conflating the preterminal tags I discard the morphological information required by the lexical macros in the f-structure annotation algorithm to project this information to the level of f-structure.

6.3.2 Final Results

In order to optimise both coverage and f-structure quality, I refined the morphological masking process to include a subsequent unmasking step so as to correctly trigger the lexical macros. The masking-unmasking process works as follows. The trees in the treebank are transformed in two ways: the lemmas are removed leaving behind the surface forms of the words and the preterminal tags are conflated to more general POS tags. The masked information is not disposed of but stored in a tab-delimited data file in the following format: full preterminal tag, surface form of word, lemma. For example: `vaip3s0 ha haber`. The grammars are extracted from the preprocessed morphologically masked trees and used to parse new text as before. The trees produced by the parser then go through a new postprocessing unmasking stage. The lemma information is reinserted and the conflated tags are expanded. Next the lexical macros are triggered by the now fully unmasked POS tags

	PCFG	A-PCFG	P-PCFG	PA-PCFG	Bikel
Parses (out of 336)	334	330	305	264	328
Labelled F-Score	79.01	78.89	78.78	78.44	79.19
Unlabelled F-Score	82.64	82.45	82.61	81.86	82.28
Fragmentation (336 F-Structures)	96.11	93.64	85.90	71.21	88.41
All GFs (100 F-Structures)	79.53	77.76	74.00	62.01	79.85
Preds-Only (100 F-Structures)	69.41	68.01	66.02	55.88	73.20

Table 6.8: Final Parsing Results for Spanish

and all f-structure equations are sent to the constraint solver as before. The f-structures produced now contain morphological information. The results are shown in Table 6.8. As expected, the `evalb` and fragmentation results are unchanged. When compared to the initial f-structure results in Table 6.7, the improvement in the all-grammatical-functions due to this extra step is clear: between 15% and 20% for all of the grammars. There are also slight improvements for the preds-only scores of the PCFG and Bikel. The extended Bikel parsing engine performs best overall: all-grammatical-functions (79.85%) and preds-only (73.20%). The PCFG, A-PCFG and P-PCFG produce f-structures of roughly similar quality. The results reported for the PA-PCFG are considerably lower. There is a general trend that the more fine-grained the grammar, the worse the coverage with PA-PCFG achieving only 71.21% fragmentation. This reflects data-sparseness problems due to the comparatively small data set. In contrast to English (Johnson, 1999), for Spanish the parent transformation has an adverse effect on parse quality.

6.4 Lexical Extraction

The method for automatically inducing semantic forms presented in this thesis is highly suitable for multilingual lexical extraction as it works on the level of the more language-independent f-structure rather than the more language-dependent c-structure. The extraction algorithm originally developed for English can be applied with no further refinement to the set of f-structures automatically generated from the Cast3LB in order to induce lexical resources for Spanish. I automatically extract 4090 non-empty semantic forms. As for English, I associate conditional probabilities with the extracted frames, differentiate between active and passive frames, parameterise frames with obliques for specific prepositions and optionally include details of syntactic category. Unlike English, the Spanish frames do not yet reflect long distance dependencies (LDDs). Of these extracted frames, 3136 are for 1401 verbal lemmas, i.e. 2.4 semantic forms per verb. The verbal semantic forms display all 98 of the frame types extracted. Table 6.9 provides an overview of the main extraction results broken down by category.

Table 6.10 shows the most frequently occurring semantic forms extracted from the Cast3LB Treebank. The most frequent frame for the verb *haber* (auxiliary 'have') is `haber[obj]` due to the Spanish construction with a invariant form of this verb (*hay*) meaning 'there is' or 'there are' which never occurs with an overt subject. Table 6.11 shows the attested semantic forms for the verb *ver* ('see') with their associated conditional probabilities. Note that as for English, the passive frame is marked with p. The passive is realised in three ways in Spanish. The verb 'to be' (*ser*) is combined with a past participle in a manner similar to the English construction. Consider Figure 6.1 where the string *ha sido exigido* can be translated word for word to the English 'has been demanded'. The annotation algorithm uses left-right context information to annotate *sido* with the f-structure equation ↑PASSIVE=+ which is exploited by the lexical extraction algorithm at f-structure level. A reflexive construction may also be used to express the passive. For example, ... *se registró un descenso...* ('... a descent was registered...') where *un descenso* is the surface subject of the normally transitive *registrar*. In Cast3LB the pronominal constituent (*se*) is tagged as a `morfema.verbal` and has an additional functional tag - PASS which is used by the annotation algorithm to assign the ↑PASSIVE=+ f-structure equation. Finally, the Spanish passive may be realised using the third person plural of the verb to be passivised with an empty subject. In this case the verb used passively will not be marked as such because it does not display the movement typically associated with the passive and is essentially an active construction with an empty subject.

6.5 Further Multilingual Grammar and Lexicon Extraction

Chapter 6 focuses on the automatic annotation of the Spanish Cast3LB treebank for the automatic extraction of LFG-based grammatical and lexical resources. In this section I briefly summarise the porting of the annotation and extraction methodologies to German

	Semantic Form Types	Lemmas	Frame Types
Total	4090	2322	98
Verbal	3136	1401	98
Nominal	432	432	3
Adverbial	26	24	4
Adjectival	496	474	20

Table 6.9: Spanish Semantic Forms broken down by Category

Semantic Form	**Frequency**
`ser([subj,xcomp])`	1202
`estar([subj,xcomp])`	208
`tener([subj,obj])`	206
`poder([subj,xcomp])`	135
`haber([obj])`	109

Table 6.10: The most frequently occurring Semantic Forms extracted from Cast3LB

Semantic Form	**Conditional Probability**
`ver([subj,obj])`	0.468
`ver([subj])`	0.290
`ver([subj,comp])`	0.121
`ver([subj],p)`	0.072

Table 6.11: Automatically extracted Lexical Entries for *ver* (see) with Associated Conditional Probabilities

and Chinese with particular emphasis on the lexical extraction work carried out for those languages.

6.5.1 German

(Cahill et al., 2003) and (Cahill et al., 2005b) describe the porting of the annotation algorithm as well as the grammar and lexicon extraction methodologies to German. As for Spanish, the two major issues which had to be addressed were the typological differences between English and German and the differences in data-structure encoding in the two treebanks. The TIGER Treebank (Brants et al., 2002) is a corpus of approximately 40,000 syntactically annotated German newspaper sentences. The annotations consist of generalised graphs, which may contain crossing and secondary edges representing LDDs and other reentrancies. Like Spanish, German configurational information is not an indicator of grammatical function and so cannot be exploited as it was in English. The TIGER graphs are converted to trees and automatically annotated using the original English f-structure annotation algorithm architecture seeded with German linguistic information and TIGER-specific encodings. Probabilistic LFG approximations for German are extracted and achieve 71.0% f-score (preds-only) against a manually constructed TIGER gold standard. A German lexical resource was automatically extracted from the f-structure-annotated treebank using the same methodology described for English and Spanish. In total, we extracted 8632 non-empty semantic forms types, 7081 of which are for 4331 verb lemmas, i.e. 1.63 semantic forms per verb when the OBL functions are parameterised for specific prepositions. Table 6.12 shows the numbers of distinct frame types extracted when `pred` values are ignored. The effects of thresholding are also shown. Table 6.13 shows the attested semantic forms for the verb *aufhören* ('stop') with their associated conditional probabilities. The coverage of an induced lexicon on new data gives a measure of the rate of acquisition of lexical information. To estimate this rate, we extract a reference lexicon from trees 1-8000 and 10001-40020 of the TIGER Treebank and a test lexicon from trees 8001-10000 and compare them. Table 6.14 shows the results of the evaluation of the coverage of an induced lexicon for verbs only. 86.75% of the semantic form types extracted from trees 8001-10000 also occur in the reference lexicon. For the

	# Sem Forms with distinct obls
# Frame Types	189
# Singletons	91
# Twice Occurring	23
# Occurring max. 5	130
# Occurring > 5	59

Table 6.12: Number of Distinct Frames (including Specific Preposition) for German Verbs Extracted from TIGER

Semantic Form	Occurrences	Conditional Probability
aufhören([subj])	8	0.444
aufhören([subj,xcomp])	7	0.389
aufhören([subj,obl:mit])	2	0.111
aufhören([comp,subj])	1	0.056

Table 6.13: Semantic Forms with Associated Conditional Probability for the Verb aufhören

13.25% which are unseen, we can distinguish between known words which have an entry in the reference lexicon (but without the relevant frame) and unknown words which do not exist at all in the reference lexicon. Similarly we make the distinction between known and unknown frames. There were no unknown frames. Table 6.14 shows that 7% of the unknown entries are new verb-frame combinations where both the verb and the frame have occurred in the reference lexicon.[3]

6.5.2 Chinese

(Burke et al., 2004c) describe the migration of the annotation and extraction techniques to the Penn Chinese Treebank (CTB) (Xue et al., 2002) which consists of of 4185 sentences of Xinhua newswire text in Mandarin Chinese. The CTB assumes that Chinese is configurational with annotation similar to the Penn-II. As for the other languages the annotation algorithm is seeded with Chinese linguistic information to produce a version of the treebank annotated with f-structure equations. A number of probabilistic LFG approximations are extracted from the treebank and used to parse new text. We achieve a preds-only f-score of 70.5% on a manually constructed gold standard for 50 trees randomly

[3] I did not carry out similar experiments for the Chinese and Spanish Treebanks due to their smaller size.

Entries also in reference lexicon:	86.75%
Entries not in reference lexicon:	13.25%
Known words:	7.00%
- Known words, known frames:	7.00%
- Known words, unknown frames:	-
Unknown words:	6.25%
- Unknown words, known frames:	6.25%
- Unknown words, unknown frames:	-

Table 6.14: Coverage of Induced German Lexicon on Unseen Data (Verbs Only)

	Semantic Form Types	Frame Types
All forms	10469	26
Verbal	2510	26
Nominal	6227	4
Adjectival	715	1
Adverbial	579	1

Table 6.15: Chinese Semantic Forms extracted from CTB broken down by Category

selected from the CTB. We extract a total of 10479 semantic forms types with 26 distinct frames types from the f-structure-annotated CTB. Of these 2510 are verbal semantic form types which occur with all 26 distinct frame types. The semantic forms extracted are broken down by category in Table 6.15.

6.6 Summary

In this chapter, I showed how the methodology for automatically annotating the Penn-II Treebank with LFG f-structure equations for the purpose of extracting grammatical and lexical resources can be adapted to Spanish. The methodology has also been successfully migrated to German and Chinese. Our methodology constitutes a novel approach to deep multilingual constraint-based grammar and lexical acquisition based on treebank resouces and automatic f-structure annotation algorithms. As treebanks become available for a growing number of languages, we expect our method to be able to deliver robust, wide-coverage multilingual resources with a substantial reduction in development cost compared to hand-crafting grammatical and lexical resources. The multilingual work presented here is very much proof-of-concept. For each language (Spanish, German and Chinese) just three months of development effort have been invested to induce the resources and further work is required to integrate LDD resolution and to refine the grammar and lexicon extraction. Once more mature multilingual induced resources are available, we expect to use them also in parser-based (rather than treebank-based) lexicon induction as described in Chapter 6.

This chapter focused on applying our methodology to the Spanish Cast3LB Treebank. I developed and applied an automatic f-structure annotation algorithm to the treebank and measured its coverage as well as the quality of the annotations. Over 96% of the trees in the treebank receive one covering and connected f-structure. When evaluated against a gold standard of 100 hand-crafted f-structures, the algorithm scores over 95% for preds-only and all-grammatical-functions. I extract four different PCFGs from the treebank and use them to parse 336 sentences set aside for testing. I also extend and retrain Bikel's statistical parsing engine with a Spanish language package to parse the test set. The retrained Bikel parser performs best against the gold standard achieving a preds-only f-score of 73.20% against the gold standard. I extract 4090 non-empty semantic forms from the annotated treebank using the same methodology applied to the Penn-II Treebank. Long-distance dependency resolution, refinement and extension of the annotation algorithm, grammar and lexicon extraction as well as the evaluation of the lexical resources remain as future work.

Chapter 7

Conclusions

Manually developing large-scale lexical resources is knowledge-intensive, time-consuming and expensive. This thesis focuses on the rapid induction of rich, large-scale lexical resources using an automatic f-structure annotation algorithm and constraint-based parsing techniques.

This thesis has:

- presented a basic methodology for the extraction of lexical resources using an automatic f-structure annotation algorithm to enrich basic syntactic trees with f-structure information (Cahill et al., 2002a, 2004a; Burke et al., 2004b).
- applied the basic methodology to the Penn-III Treebank (75,000 sentences).
- developed a mapping between the extracted semantic forms and the manually constructed OALD and COMLEX resources for the purposes of large-scale evaluation.
- examined the coverage of the lexicon induced from the treebank on new text and illustrated the rate of accession of lexical information.
- reported on a task-based evaluation of the extracted resources by incorporating them into the parsing architectures of Cahill et al. (2004b) and demonstrating the improvement in the quality of parser output as a result.
- extended the basic treebank-based methodology to create a modular and flexible architecture capable of processing raw text by incorporating taggers such as MXPOST

(Ratnaparkhi, 1996) and TreeTagger (Schmid, 1994) and history-based parsers such as (Charniak, 2000), (Collins, 1999) (Models 2 and 3) and (Bikel, 2002).

- compared treebank- and parser-based lexical extraction.
- applied the extended architecture to the 65,000 word test set of Korhonen (2002), to evaluate the extraction system against hers.
- applied the extended architecture to 90 million words of the BNC and evaluated the output against COMLEX and the OALD employing the mapping mechanism used for the treebank-induced resources.
- examined the effect of sentence length on the quality and coverage of the induced lexicons.
- computed rates of lexical accession over the larger BNC corpus.
- applied and evaluated the annotation and grammar and lexicon extraction techniques to Spanish and the Cast3LB Treebank.

The methodology at the heart of this thesis differs from other approaches to automatic lexical acquisition in that basic syntactic patterns are not read off directly from trees. Instead the trees (from treebanks or parsers) are first enriched automatically with f-structure information relying on linguistic generalisations encoded in the automatic f-structure annotation algorithm. The result is a richer set of semantic forms which additionally capture LDD information in the source data structure. Methods most similar to this are (Hockenmaier et al., 2004) for CCG and (Miyao et al., 2004) for HPSG. Unlike our approach, both of these rely on a preprocessing step which cleans up inconsistencies or awkward constructions in the treebank. In addition, neither of these approaches have carried out extensive qualitative evaluations against external machine-readable lexical resources. Also in contrast to these approaches, we have acquired large-scale lexical resources from 90 million words of BNC text. To extract as accurate a resource as possible, we modify the LDD resolution approach of (Cahill et al., 2004b), feeding a restricted set of 'safe' semantic forms back into the LDD resolution component before extracting a final complete set of all semantic forms from the BNC. The architecture is modular, allowing us to experiment with different tagger and parser combinations.

Evaluation of the induced lexical resources is an important facet of the work presented in this thesis. Our aim was to produce a large-scale, accurate lexicon. In order to verify this, it was necessary to carry out large-scale evaluations. We use two different manually constructed lexicons as gold standards, the OALD and COMLEX. Our largest evaluation is for 4703 verb lemmas, to our knowledge the most extensive such evaluation carried out for English. In the case of all evaluations for both treebank- and parser-based extraction, we exceed the baselines. There are many issues associated with evaluating against an externally produced lexicon which we have highlighted in this thesis. First, there is the issue of gold standard coverage. Our detailed error analysis pointed to the fact that the selected gold-standard lexicons were not always complete or accurate. While we found it useful to be able to evaluate frames parameterised for prepositions and particles against COMLEX, it became obvious that COMLEX tends to over-generate in its assignment of prepositional phrases to verbs, in particular a list of 30 directional prepositions is associated in its entirety with certain verbs. There were also issues with the OALD resource in terms of coverage. In addition, the extracted frames were difficult to map to the OALD as there was not always as clear an argument/adjunct distinction in the OALD as in COMLEX. Extending our methodology to deal with raw text afforded us the opportunity of directly comparing our extraction procedure to the best results of Korhonen (2002), thanks to her freely downloadable evaluation resources. For a test set of 29 verbs, our system significantly outperformed hers. The automatic acquisition of LFG lexical resources is part of a larger project for the acquisition of constraint-based linguistic resources. The extracted semantic forms (both active and passive) along with their associated conditional probabilities are used in the LDD resolution component of the parsing architecture presented in Cahill et al. (2004b). In order to measure the usefulness of the resources, we present an evaluation of the parser output with and without the incorporation of semantic forms for LDD resolution. We show that the resolution step improves parser performance by up to 5.31% against the DCU-105 gold standard.

For both the treebank- and parser-based extraction experiments, we carried out qualitative experiments to examine the coverage of the extracted lexicon on unseen text. We found that even in a limited language domain such as the Penn-II WSJ section, a reference lexicon extracted from Sections 02-21 had just under 90% coverage of a test lexicon extracted from Section 23. The majority of the unseen entries were a new combination of a previously seen frame with a previously seen verb lemma. Using a test lexicon extracted from the domain diverse Brown Corpus saw the reference lexicon coverage drop further to 65.37% (occurences greater than one). In this case the problem is with unseen verb lemmas. Almost 20% of the lexicon is made up of entries comprising an unseen lemma in combination with a previously seen frame. Plotting the accession rates for semantic form types, lemmas and frames types in the BNC data displayed a similar propensity.

The methodology developed in this thesis has been ported to other typologically diverse languages: German (Cahill et al., 2003, 2005b), Chinese (Burke et al., 2004c) and Spanish (O'Donovan et al., 2005b). In this thesis, I focus on the work carried out for Spanish using the Cast3LB Treebank (Civit, 2003). With just three months of development effort, I have created high-quality LFG grammatical and lexical resources for Spanish. I extend Bikel's (2002) Multilingual Parsing Engine to include a Spanish language module and retrain it on the Cast3LB. Incorporating this as c-structure engine into the pipeline parsing model of Cahill et al. (2004b) we achieve an f-score of 73.20% (preds-only) against a manually constructed set of 100 gold standard f-structures. In addition, we extract 3136 semantic forms for 1401 verb lemmas.

7.1 Future Work

There are many avenues of further research stemming from the work presented here. Improving the quality of the induced lexicons is of course a constant aim. It is important therefore to carry out further evaluation of the acquired lexicons. Merging COMLEX and the OALD to create a more comprehensive resource for dictionary-based evaluation would be very worthwhile. It is also hoped that the conditional probabilities of the extracted semantic forms will be evaluated using measures of distributional similarity and rank accuracy.

The focus of this thesis has been on using high-quality annotation and parsing techniques, along with simple statistical filtering to produce high-quality lexical resources. The statistical parsers used are trained on WSJ text which potentially limits their effectiveness on more domain-diverse data such as the BNC (Gildea, 2001). Using more varied training data (e.g. the parse-annotated Brown subcorpus) could prove valuable.

Although Korhonen et al. (2000) show that the use of relative frequencies to filter extracted subcategorisation frames is more effective than both the Binomial Hypothesis Test and the Log Likelihood Ratio, it would be worthwhile to investigate the effect of using more refined statistical filtering in the extraction system.

At present, a resource such as COMLEX is of limited use for the purpose of guiding probabilistic parsers, dismabiguation and resolving LDDs as it does not contain conditional probabilities required for resolution. It could be interesting, however, to use COMLEX to bootstrap lexical acquisition: only accept verb/frame combinations which occur in COMLEX and calculate their relative frequency.

Finally, as regards multilingual lexical extraction, there is still a lot to do. The Chinese, German and Spanish lexicons have yet to be evaluated. The annotation algorithm for Spanish requires further work to improve coverage and quality. The parsing coverage and results need to be improved, specifically when the automatically produced f-structures fail to contain empty subjects resulting from pro-drop constructions. Incorporating the Spanish lexical resources in an LDD resolution component also remains as future work.

Bibliography

Ades, A. and Steedman, M. (1982). On the Order of Words. *Linguistics and Philosophy*, 4:517–558.

Bernard, L. (2002). *User Reference Guide for the British National Corpus.* Technical Report, Oxford University Computing Services.

Bikel, D. M. (2002). Design of a Multi-lingual, Parallel-processing Statistical Parsing Engine. In *Proceedings of the Human Language Technology Conference*, pages 24–27, San Diego, CA.

Boguraev, B., Briscoe, E., Carroll, J., Carter, D., and Grover, C. (1987). The Derivation of Grammatically Indexed Lexicon from the Longman Dictionary of Contemporary English. In *Proceedings of the 25th Annual Meeting of the Association of Computational Linguistics*, pages 193–200, Stanford, CA.

Brants, T., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER Treebank. In Hinrichs, E. and Simov, K., editors, *Proceedings of the first Workshop on Treebanks and Linguistic Theories (TLT'02)*, pages 24–41, Sozopol, Bulgaria.

Brent, M. (1993). From Grammar to Lexicon: Unsupervised Learning of Lexical Syntax. *Computational Linguistics*, 19(2):203–222.

Bresnan, J. (2001). *Lexical-Functional Syntax.* Blackwell, Oxford.

Briscoe, E. (2001). From dictionary to corpus to self-organizing dictionary: learning valency associations in the face of variation and change. In *Proceedings of Corpus Linguistics 2001*, pages 79–89, Lancaster, UK.

Briscoe, E. and Carroll, J. (1997). Automatic Extraction of Subcategorization from Corpora. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing*, pages 356–363, Washington, DC.

Burke, M. (2006). *Automatic Annotation of the Penn-II Treebank with F-Structure Information.* PhD thesis, School of Computing, Dublin City University, Dublin, Ireland. Forthcoming.

Burke, M., Cahill, A., O'Donovan, R., van Genabith, J., and Way, A. (2004a). The Evaluation of an Automatic Annotation Algorithm against the PARC 700 Dependency Bank. In *Proceedings of the Ninth International Conference on LFG*, pages 101–121, Christchurch, New Zealand.

Burke, M., Cahill, A., O'Donovan, R., van Genabith, J., and Way, A. (2004b). Treebank-Based Acquisistion of Wide-Coverage, Probabilistic LFG Resources: Project Overview, Results and Evaluation. In *The First International Joint Conference on Natural Language Processing (IJCNLP-04), Workshop "Beyond shallow analyses - Formalisms and statistical modeling for deep analyses"*, page [no page numbers], Hainan Island, China.

Burke, M., Lam, O., Chan, R., Cahill, A., O'Donovan, R., Bodomo, A., van Genabith, J., and Way, A. (2004c). Treebank-Based Acquisition of a Chinese Lexical-Functional Grammar. In *Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, pages 161–172, Tokyo, Japan.

Butt, M., Dyvik, H., King, T. H., Masuichi, H., and Rohrer, C. (2002). The Parallel Grammar Project. In *Proceedings of COLING 2002, Workshop on Grammar Engineering and Evaluation*, pages 1–7, Taipei, Taiwan.

Butt, M., King, T. H., Niño, M. E., and Segond, F. (1999). *A Grammar Writer's Cookbook.* CSLI Publications, Stanford, CA.

Cahill, A., Burke, M., McCarthy, M., O'Donovan, R., van Genabith, J., and Way, A. (2004a). Evaluating Automatic F-Structure Annotation for the Penn-II Treebank. In Erhard Hinrichs and Kiril Simov, editor, *Journal of Language and Computation; Special*

Issue on "Treebanks and Linguistic Theories", pages 523–547, Dordrecht, The Netherlands. Kluwer Academic Press.

Cahill, A., Burke, M., O'Donovan, R., Riezler, S., van Genabith, J., and Way, A. (2005a). Shallow and Deep Parser Comparison with Automatically Generated Dependency Relations. *Computational Linguistics*, (Under Review).

Cahill, A., Burke, M., O'Donovan, R., van Genabith, J., and Way, A. (2004b). Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 320–327, Barcelona, Spain.

Cahill, A., Forst, M., Burke, M., McCarthy, M., O'Donovan, R., Rohrer, C., van Genabith, J., and Way, A. (2005b). Treebank-Based Acquisition of Multilingual Unification Grammar Resources. In Emily Bender, Dan Flickinger, Frederik Fouvry and Melanie Siegel, editor, *Journal of Research on Language and Computation; Special Issue on "Shared Representations in Multilingual Grammar Engineering"*, Dordrecht, The Netherlands. Kluwer Academic Press (to appear).

Cahill, A., Forst, M., McCarthy, M., O'Donovan, R., Rohrer, C., van Genabith, J., and Way, A. (2003). Treebank-Based Multilingual Unification-Grammar Development. In *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development, at the 15th European Summer School in Logic Language and Information*, pages 17–24, Vienna, Austria.

Cahill, A., McCarthy, M., van Genabith, J., and Way, A. (2002a). Automatic Annotation of the Penn Treebank with LFG F-Structure Information. In *Proceedings of the LREC Workshop on Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data*, pages 8–15, Las Palmas, Canary Islands, Spain.

Cahill, A., McCarthy, M., van Genabith, J., and Way, A. (2002b). Parsing with PCFGs and Automatic F-Structure Annotation. In Butt, M. and King, T. H., editors, *Proceedings of the Seventh International Conference on Lexical Functional Grammar*, pages 76–95, Stanford, CA. CSLI Publications.

Carroll, G. and Rooth, M. (1998). Valence Induction with a Head-Lexicalised PCFG. In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing*, pages 36–45, Granada, Spain.

Carroll, J. and Briscoe, E. (2002). High precision extraction of grammatical relations. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 134–140, Taipei, Taiwan.

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

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

Charniak, E. (1996). Tree-Bank Grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1031–1036, Menlo Park, CA.

Charniak, E. (2000). A maximum entropy inspired parser. In *Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000)*, pages 132–139, Seattle, WA.

Chen, J. and Vijay-Shanker, K. (2000). Automated Extraction of TAGs from the Penn Treebank. In *Proceedings of the 38th Annual Meeting of the Association of Computational Linguistics*, pages 65–76, Hong Kong.

Civit, M. (2003). *Criterios de etiquación y desambiguación morfosintáctica de corpus en español.* PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain.

Collins, M. (1997). Three Generative, Lexicalized Models for Statistical Parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain.

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

Crouch, R., Kaplan, R., King, T. H., and Riezler, S. (2002). A comparison of evaluation metrics for a broad coverage parser. In *Proceedings of the LREC Workshop: Beyond PARSEVAL – Towards Improved Evaluation Measures for Parsing Systems*, pages 67–74, Las Palmas, Canary Islands, Spain.

Dalrymple, M. (2001). *Lexical-Functional Grammar.* San Diego, CA; London. Academic Press.

Dudenredaktion, editor (2001). *DUDEN – Das Stilworterbuch. Number 2 in Duden in zwölf Banden.* Dudenverlag, Mannheim, Germany.

Frank, A., Sadler, L., van Genabith, J., and Way, A. (2003). From Treebank Resources to LFG F-Structures. In Abeillé, A., editor, *Treebanks: Building and Using Syntactically Annotated Corpora*, pages 367–389. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Gazdar, G. and Mellish, C. (1989). *Natural Language Processing in PROLOG: An Introduction to Computational Linguistics.* Addison-Wesley Publishing Co., Wokingham, England.

Gildea, D. (2001). Corpus Variation and Parser Performance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 167–172, Pittsburgh, PA.

Grishman, R., MacLeod, C., and Meyers, A. (1994). Comlex Syntax: Building a Computational Lexicon. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*, pages 268–272, Kyoto, Japan.

Hajic, J. (1998). Building a syntactically annotated corpus: The Prague Dependency Treebank. In *Issues in Valency and Meaning*, pages 106–132. Karolinum, Prague, Czech Republic.

Hockenmaier, J. (2003). Parsing with Generative Models of Predicate-Argument Structure. In *Proceedings of the 41st Annual Conference of the Association for Computational Linguistics*, pages 359–366, Sapporo, Japan.

Hockenmaier, J., Bierner, G., and Baldridge, J. (2004). Extending the Coverage of a CCG System. *Language and Computation*, **2**(2):165–208.

Hockenmaier, J. and Steedman, M. (2002). Acquiring Compact Lexicalized Grammars from a Cleaner Treebank. In *Proceedings of Third International Conference on Language Resources and Evaluation*, pages 1974–1981, Las Palmas, Spain.

Hornby, A., editor (1980). *Oxford Advanced Learner's Dictionary of Current English.* Oxford University Press, Oxford, UK.

Johnson, M. (1999). PCFG models of linguistic tree representations. *Computational Linguistics*, **24**(4):613–632.

Joshi, A. (1988). Tree Adjoining Grammars. In Dowty, D., Karttunnen, L., and Zwicky, A., editors, *Natural Language Parsing*, pages 206–250. Cambridge University Press, Cambridge, UK.

Kaplan, R. and Bresnan, J. (1982). Lexical Functional Grammar, a Formal System for Grammatical Representation. In Bresnan, J., editor, *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, Cambridge, MA.

King, T. H., Crouch, R., Riezler, S., Dalrymple, M., and Kaplan, R. (2003). The PARC 700 Dependency Bank. In *Proceedings of the EACL03: 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*, pages 1–8, Budapest, Hungary.

Kinyon, A. and Prolo, C. (2002). Identifying Verb Arguments and their Syntactic Function in the Penn Treebank. In *Proceedings of the 3rd LREC Conference*, pages 1982–1987, Las Palmas, Canary Islands, Spain.

Korhonen, A. (1998). Automatic Extraction of Subcategorization Frames from Corpora - Improving Filtering with Diathesis Alternations. In *Proceedings of the ESSLLI 98 Workshop on Automated Acquisition of Syntax and Parsing*, pages 49–56, Saarbrücken, Germany.

Korhonen, A. (2002). *Subcategorization Acquisition.* PhD thesis, Computer Laboratory, University of Cambridge, UK.

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*, pages 199–205, Hong Kong.

Krotov, A., Hepple, M., Gaizauskas, R. J., and Wilks, Y. (1998). Compacting the Penn Treebank Grammar. In *Proceedings of COLING/ACL98: Joint Meeting of the 36th Annual Meeting of the ACL and the 17th International Conference on Computational Linguistics*, pages 699–703, Montréal, Canada.

Lapata, M. (1999). Acquiring lexical generalizations from corpora: A case study for diathesis alternations. In *Proceedings of the 37th Meeting of the Association of Computational Linguistics (ACL)*, pages 397–404, College Park, MD.

Levin, B. (1993). *English Verb Classes and Alternations.* Chicago University Press, Chicago, IL.

Macleod, C., Meyers, A., and Grishman, R. (1994). The COMLEX Syntax Project: The First Year. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 669–703, Princeton, NJ.

Magerman, D. (1994). *Natural Language Parsing as Statistical Pattern Recognition.* PhD thesis, Department of Computer Science, Stanford University, CA.

Magerman, D. (1995). Statistical decision-tree models for parsing. In *Proceedings of the 33rd Meeting of the Association for Computational Linguistics*, pages 276–283, Cambridge, MA.

Manning, C. (1993). Automatic Acquisition of a Large Subcategorisation Dictionary from Corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 235–242, Columbus, OH.

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. (1994). The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 110–115, Princeton, NJ.

Marinov, S. (2004). Automatic Extraction of Subcategorization Frames for Bulgarian. In *Proceedings of 16th ESSLLI Student Session*, Nancy, France.

McCarthy, D. and Korhonen, A. (1998). Detecting Verbal Participation in Diathesis Alternations. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 1493–1495, Montreal, Canada.

McCarthy, M. (2003). *Design and Evaluation of the Linguistic Basis of an Automatic F-Structure Annotation Algorithm for the Penn-II Treebank.* Master's thesis, School of Computing, Dublin City University, Dublin, Ireland.

Meyers, A., MacLeod, C., and Grishman, R., editors (1994). *Standardization of the Complement Adjunct Distinction.* New York University, New York.

Miyao, Y., Ninomiya, T., and Tsujii, J. (2003). Probabilistic modeling of argument structures including non-local dependencies. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 285–291, Borovets, Bulgaria.

Miyao, Y., Ninomiya, T., and Tsujii, J. (2004). Corpus-oriented Grammar Development for Acquiring a Head-Driven Phrase Structure Grammar from the Penn Treebank. In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, page [no page numbers], Hainan Island, China.

Nakanishi, H., Miyao, Y., and Tsujii, J. (2004). Using Inverse Lexical Rules to Acquire a Wide-coverage Lexicalized Grammar. In *Proceedings of the Workshop 'Beyond Shallow Analyses - Formalisms and Statistical Modelling for Deep Analyses' at the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, page [no page numbers], Hainan Island, China.

Noreen, E. W. (1989). *Computer Intensive Methods for Testing Hypotheses: An Introduction.* Wiley, New York.

O'Donovan, R., Burke, M., Cahill, A., van Genabith, J., and Way, A. (2004). Large-Scale Induction and Evaluation of Lexical Resources from the Penn-II Treebank. In *Pro-*

ceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, pages 368–375, Barcelona, Spain.

O'Donovan, R., Burke, M., Cahill, A., van Genabith, J., and Way, A. (2005a). Large-Scale Induction and Evaluation of Lexical Resources from the Penn-II and Penn-III Treebanks. *Computational Linguistics*, 31(3):329–365.

O'Donovan, R., Cahill, A., van Genabith, J., and Way, A. (2005b). Automatic Acquisition of Spanish LFG Resources from the Cast3LB Treebank. In *Proceedings of the Tenth International Lexical Functional Grammar Conference*, Bergen, Norway, to appear.

Pollard, C. and Sag, I. (1994). *Head-driven Phrase Structure Grammar*. CSLI Publications, Stanford, CA.

Proctor, P., editor (1978). *Longman Dictionary of Contemporary English*. Longman, London.

Ratnaparkhi, A. (1996). A Maximum Entropy Part-of-Speech Tagger. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP) Conference*, pages 133–142, Philadelphia, PA.

Riezler, S., King, T., Kaplan, R., Crouch, R., Maxwell, J. T., and Johnson, M. (2002). Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*, pages 271–278, Philadelphia, PA.

Roland, D. and Jurafsky, D. (1998). How Verb Subcategorization 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*, pages 1117–1121, Montreal, Canada.

Roland, D., Jurafsky, D., Menn, L., Gahl, S., Elder, E., and Riddoch, C. (2000). Verb Subcategorization Frequency Differences between Business News and Balanced Corpora: The Role of Sense. In *Proceedings of the Association for Computational Linguistics (ACL-2000) Workshop on Comparing Corpora*, pages 28–34, Hong Kong, China.

Sadler, L., van Genabith, J., and Way, A. (2000). Automatic F-Structure Annotation from the AP Treebank. In Butt, M. and King, T. H., editors, *Proceedings of the Fifth International Conference on Lexical Functional Grammar*, pages 226–243, Stanford, CA. CSLI Publications.

Sarkar, A. and Zeman, D. (2000). Automatic Extraction of Subcategorization Frames for Czech. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 691–697, Saarbrücken, Germany.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Schmid, H. (2004). Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2004)*, pages 162–168, Geneva, Switzerland.

Schulte im Walde, S. (2002a). A Subcategorisation Lexicon for German Verbs induced from a Lexicalised PCFG. In *Proceedings of the 3rd LREC Conference*, pages 1351–1357, Las Palmas, Canary Islands, Spain.

Schulte im Walde, S. (2002b). Evaluating Verb Subcategorisation Frames learned by a German Statistical Grammar against Manual Definitions in the Duden Dictionary. In *Proceedings of the 10th EURALEX International Congress*, pages 187–197, Copenhagen, Denmark.

Simov, K., Popova, G., and Osenova, P. (2002). HPSG-based syntactic treebank of Bulgarian (BulTreeBank). In Wilson, A., Rayson, P., and McEnery, T., editors, *A Rainbow of Corpora: Corpus linguistics and the languages of the world*, pages 135–142. Lincon-Europa, Munich, Germany.

Ushioda, A., Evans, D., Gibson, T., and Waibel, A. (1993). The Automatic Acquisition of Frequencies of Verb Subcategorization Frames from Tagged Corpora. In *SIGLEX ACL Workshop on the Acquisition of Lexical Knowledge from Text*, pages 95–106, Columbus, OH.

van Genabith, J., Sadler, L., and Way, A. (1999a). Data-driven Compilation of LFG Semantic Forms. In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora (LINC-99)*, pages 69–76, Bergen, Norway.

van Genabith, J., Sadler, L., and Way, A. (1999b). Semi-automatic Generation of F-Structures from Treebanks. In Butt, M. and King, T., editors, *Proceedings of the Fourth International Conference on Lexical Functional Grammar*, Manchester, UK.

Xia, F. (1999). Extracting Tree Adjoining Grammars from Bracketed Corpora. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium(NLPRS-99)*, pages 398–403, Beijing, China.

Xue, N., Chiou, F.-D., and Palmer, M. (2002). Building a Large-Scale Annotated Chinese Corpus. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 1–8, Taipei, Taiwan.

Appendix A

Spanish Head Rules

Mother Category	Direction	Ordered Head Candidates
S	l	gv S
S.F.C	l	gv S.F.C
S.F.R	l	gv S.F.R
S.F.A	l	gv S.F.A
S.F.AComp	l	gv S.F.AComp
S.F.ACond	l	gv S.F.ACond
S.F.AConc	l	gv S.F.AConc
S.F.ACons	l	gv S.F.ACons
S.NF.C	l	infinitiu S.NF.C
S.NF.P	l	ao aq S.NF.P
S.NF.PA	l	gv vmi vms S.NF.PA
S.NF.A	l	gerundi S.NF.A
S.NF.R	l	infinitiu gv S.NF.R
S.NF.C*	l	infinitiu S.NF.C*
S.NF.P*	l	ao aq S.NF.P*
S.NF.PA*	l	gv vmi vms S.NF.PA*
S.NF.A*	l	gerundi S.NF.A*
S.NF.R*	l	infinitiu gv S.NF.R*
S*	l	gv S* S.F.AConc sn sadv

S.F.C*	l	gv S.F.C* sp
S.F.R*	l	gv S.F.R*
S.F.A*	l	gv S.F.A*
S.F.AComp*	l	gv S.F.AComp* prep
S.F.ACond*	l	gv S.F.ACond*
S.F.AConc*	l	gv S.F.AConc*
S.F.ACons*	l	gv S.F.ACons*
sp	l	prep sps spc sp
sn	l	grup.nom relatiu sn nc np espec S.F.R S.NF.C
s.a	l	ao aq n s.a
sa	l	ao aq sa
sadv	l	rg rn pp pn pt pd pi pr px pe p0
gv	l	vmi vms vsi vss vai vas gerundi infinitiu
grup.nom	l	nc np pp pn pt pi pr px pe p0 pd Zm Zp w ao aq dd da dt dp di dn grup.nom S.NF.P
infinitiu	l	vmn vsn van infinitiu
gerundi	l	vmg vsg vag
espec	r	dd da dt dp di dn Zm z
conj.subord	l	prep cs
interjeccio	l	i I
INC	l	S S* sn S.F.A S.NF.C
neg	l	rn
prep	r	sps spc sp

Appendix B

Spanish Gold Standard Sentences

1. En 1708 era en las tres dimensiones del cuerpo propio donde ponía el primer fundamento de la distinción de las regiones en el espacio .

2. Los compuestos así obtenidos presentan , además_de propiedades semiconductoras en_ausencia_de oxidación externa - - dopado - - , una elevada estabilidad térmica que abre un gran campo de estudio de nuevos materiales con propiedades conductoras y semiconductoras .

3. Entre los métodos indirectos más insidiosos figuraría una truculenta idea con la que ya se especulaba en los años sesenta : - Sería posible , mediante el lanzamiento y diseminación de los agentes químicos apropiados , abrir un agujero en la capa de ozono sobre una nación enemiga ?

4. Los sabios en ciencias cósmicas dicen que es posible que haya otros mundos donde sociedades humanas como la_nuestra hayan vencido a la muerte y sean inmortales .

5. Aunque quizá ya no lograra reponerse .

6. Este descubrimiento clave fue origen de un gran esfuerzo investigador , que provocó la búsqueda de nuevos complejos orgánicos de transferencia de carga - CTC - que pudieran igualar o incluso mejorar al TCNQ-TTF .

7. Su pertenencia a varias comisiones parlamentarias da una idea de su capacidad de trabajo .

8. Más_de la mitad de los 100 senadores han dicho que votarán en_favor_del proyecto que garantizará a los productos chinos los mismos gravámenes que se aplican a los de otros países .

9. Toledo manifestó ante la multitud que reunió en la Plaza_de_Armas de Cañete que esas elecciones son "ilegítimas" mientras_que la gente coreaba "queremos nuevas elecciones" , "no al fraude" , y otras frases de rechazo al régimen de Fujimori .

10. En los Estados_Unidos se especula mucho con la presunta homosexualidad de Gere y hasta circulan anécdotas picantes que la confirmarían .

11. Algo parecido nos sucede a todos .

12. En marzo_de_1993 se encargó de una oficina en un pueblecito y desde hace unos meses se encuentra en la ciudad sajona .

13. Nadie duda de que Pilar hace ruido ni que tras su imagen de niña buena esconde un carácter duro .

14. Los documentos de la disolución con la firma de Mori y el sello del emperador Akihito fueron entregados al presidente de la poderosa Cámara_del_Legislativo , Soichiro_Ito , por el ministro Portavoz del gobierno Mikio_Aoki , a primera hora de la tarde .

15. Ya sé que la vida , en_general , no está para bromas y que un partido de fútbol como el de ayer , algo tan tonto y que puede ser a_la_vez algo tan bello , merece la misma calificación oficial / policial que un terremoto : máximo riesgo .

16. Campeón y caballero , aunque en el deporte la caballerosidad sea un signo de debilidad de espíritu en el seno de la fortaleza de cuerpo .

17. - Está bien , hazlo tú - le indiqué a mi mujer , e intenté salir del salón .

18. Aburrido por la falta de vida nocturna dejó la compañía y abrió una taberna en la Plaza_Mayor .

19. Entonces se felicitan con chocar de palmas y mucho alarde de festividad , y van a por la siguiente .

20. Por_otra_parte , la síntesis de nuevas moléculas aceptoras ha tenido que superar las dificultades de obtención de derivados del TCNQ , pero durante los últimos años se ha podido sintetizar un elevado número de moléculas aceptoras - fig. 5 - con una amplia gama de valores de la afinidad electrónica .

21. La pregunta que se hace el país no es ya cuántos han trincado sino cuántos no han trincado .

22. Los funerales de Obuchi que serán oficiados el día_8 de este mes en el pabellón de las artes marciales Nihon_Budokan , no revisten carácter de Estado , sino_que están patrocinados por el gobierno y el partido mayoritario Liberal_Demócrata - PLD - , que él presidió .

23. Es lo que ha hecho hasta ahora y el resultado ha sido inmejorable .

24. Antes_de 1989 el sueño de los alemanes del Este era escapar , saltar el muro , marchar a Occidente .

25. Se acabó ; se acabó todo y yo con esta depre .

26. Con el personal me llevo bien .

27. Ya sé que a Nicolás_Redondo lo quieren ante los tribunales mejor antes del 27_E que después y sé que mientras católicos y protestantes se matan en Irlanda_del_Norte , el Vaticano , siempre sensible a los problemas , discute la traducción del aún reciente Catecismo al inglés y no llega a un acuerdo sobre si Jesús_de_Nazaret es - man - o - human - .

28. En 1812 los románticos españoles inventaron en las Cortes_de_Cádiz el liberalismo político .

29. En una primera fase los etarras recogían los datos de revistas económicas y boletines especializados en industria y finanzas .

30. Hasta el momento , de Europa sólo han comunicado su asistencia Esperanza_Aguirre y el ministro del Asuntos_Exteriores de Gran_Bretaña , John_Battle , además_del viceprimer ministro ruso , Viktor_Khristenko .

31. Por qué tienen miedo de participar en una elección limpia ...

32. A ello se ha dedicado un gran esfuerzo investigador en los últimos años .

33. Sesión Continua : Seguir desaprendiendo , aunque sea con dolor , todos los dogmas que aún me habitan .

34. Y eso por no hablar de los conciertos históricos de Unión_Boricua , Los_Pleneros_de_la_21 y Los_Instantáneos_de_la_Bombiplena .

35. Claro que aún hay una tercera opción en el derribo .

36. - Me gusta este trabajo , porque aquí hay mucho por hacer .

37. - Vamos a decir que esas variedades son iguales ?

38. Pero creo que si fuese un tío feo y con barba ya lo hubiesen machacado - , comenta Sellarés .

39. ¿ Qué hago yo ahora ?

40. Pero ya nos habíamos mudado , así_que , en_lugar_de seguir buscando trabajo , pedí un crédito y organicé mi propio negocio de electricidad .

41. ¿ Qué significa tirar papeleras y quemarlas después con sus desperdicios ?

42. Y aun cuando la olvidemos , como hemos olvidado a todos los demás héroes anónimos , es gracias_a ella , y a gentes como ella , que la humanidad puede perseverar en el sueño de la felicidad y la razón .

43. De_todos_modos , vistas las pluralidades sociales y la evolución diferenciada de las diversas sociedades , - una - cultura designa un tipo particular de herencia social : según Linton la cultura en su conjunto se compone de gran número de culturas , cada una de las cuales es característica de un grupo determinado de individuos .

44. Se espera que los tres partidos de la coalición gubernamental compuesta por el Liberal_Demócrata - PLD - , el nuevo Komeito y el Nuevo_Conservador , logren la mayoría en la poderosa Cámara_Baja aunque podrían no renovar la cómoda mayoría absoluta que ahora poseen .

45. Desde fuera , el Rancho_Criollo parece un fuerte apache esperando la carga del quinto de caballería aunque el Sur_Bronx no se parece mucho al Cañón_del_Colorado : montañas de escoria y escombros , casas quemadas y coches desguazados , calles como páramos por donde no pasa un alma , tiendas desvencijadas que sirven de tapadera a camellos y maleantes .

46. Recientemente , J._Simons ha descrito un procedimiento alternativo para conseguir una disposición de macrociclos de ftalocianinas en apilamientos sin necesidad de ligandos puente .

47. Al_igual_que su colega de Leipzig , Rainer_Hanesch también marchó al Este a las órdenes de su empresa , una inmobiliaria .

48. Casi inevitablemente , ZZNM ha salido de la novela para subir a los altares y recibir el culto que merece Texto escrito para Radio_ZZPaís , sobre una conferencia pronunciada en el ZZlugar de ZZciudad , bajo el patrocinio del ZZorganización .

49. Aunque menos estudiados , los análogos de TCNQ fusionados con anillos aromáticos presentan un interés especial .

50. De_hecho , ya se han realizado experiencias con estas enfermedades a medida .

51. Quizá ya iba a quedarse así .

52. Los de andadura media , cuando la papelera no se desprende con el empellón primero , continúan pataleando y chillando , hasta que cae al suelo y se derraman todas sus especias .

53. Las cosas no han cambiado tanto , brodel - hermano - .

54. - Estábamos hartos de jardines bien cuidados con gnomos de plástico y estanques con ranas , aceras ultralimpias y árboles peinados - .

55. La Representante_Comercial de EEUU , Charlene_Barshefsky , advirtió de que una demora podría poner en peligro el acuerdo que abrirá a este país los mercados chinos con más_de 1.300_millones de consumidores .

56. ¿ No es acaso toda religión la hipóstasis del conflicto entre la inercia de este mundo material y las supremas incitaciones de otro mundo ?

57. Por_ejemplo , para algunos la cultura era la característica del hombre como ser social : en este sentido la vieja oposición entre naturaleza y espíritu llegaba a ser una separación radical , esto_es , una diferenciación absoluta entre los hombres y los animales .

58. La primera se refiere a la lengua : - Por mi parte yo veo como indicador más característico de la unidad tribal la comunidad de lenguaje , pues una tradición común de prácticas y conocimientos , de costumbres y creencias , sólo puede ser compartida por personas que utilicen la misma lengua - .

59. La corrupción es como los pasos de la procesión , aún no ha llegado a la plaza un paso cuando sale el otro dando traspiés .

60. Romero lo explica así : - Nos presentamos como una fuerza política de gobierno y siempre nos hemos presentado como una fuerza política de oposición - .

61. Como hacía buen tiempo decidió salir .

62. El que prende el fuego caníbal , desde la violencia se ha asentado ya en la destrucción , y su ritual es una catarsis salvaje .

63. Como casi todos son jóvenes , no ha habido problema en reciclarlos - .

64. El genio finalmente , - es la capacidad espiritual innata - ingenium - mediante la cual la Naturaleza da la regla al arte , pero , lejos_de ser por eso heterónoma , no cabe actividad más soberanamente libre que la de una creación genial .

65. Cuando se derrumbó el régimen de Honecker muchos lo hicieron .

66. El liberalismo tuvo cierta influencia idealista , pero la bragueta inundó el mundo .

67. El libro que leemos intenta explicarlo , pero sin exagerar Las posturas de tirios y troyanos .

68. Obuchi , de 62 años , sufrió un derrame cerebral en la madrugada del día_2_de_abril y , tras permanecer seis semanas en estado de coma , falleció en el hospital Yuntendo de Tokio .

69. Todo es nuevo , Todo está por organizar .

70. Recoge nuestros detritus , nos veta la Edad_Media , y nos hace sentir limpios , educados , pertenecientes al mundo hermoso al que , al parecer , no pertenecemos .

71. Sigue siendo del Barça y se siente irremisiblemente atraída por la lectura y por el ajedrez .

72. Son buenos trabajadores , pero en ocasiones les falta iniciativa y sentido de la responsabilidad .

73. Le hemos tomado cariño a los dos , y no tenemos corazón para premiar a uno y castigar al otro .

74. El que golpea una papelera al pasar - 2.000 destruyeron en Madrid sólo el año pasado - es un insatisfecho , que siente la tentación del punching .

75. El denominado impulso democrático parece dormir el sueño de los justos , según reconocen sus señorías de todos los grupos y colores .

76. El funcionario indicó que el número de inmigrantes indocumentados detenidos y deportados a su país en este año creció un 8_por_ciento en comparación con la cifra registrada en el mismo periodo del año_pasado .

77. El Gobierno de Estados_Unidos pidió hoy al Senado que someta a votación a_mediados_de este mes el acuerdo que otorga beneficios comerciales permanentes a China .

78. El primer ministro japonés , Yoshiro_Mori , disolvió hoy la Cámara_de_Diputados para convocar las elecciones generales el día_25_de_junio .

79. Tal exaltación no se ha efectuado sin pérdida de ciertos matices importantes del personaje ni sin la invención por hagiógrafos tardíos de virtudes muy dudosas .

80. Todos los grandes bancos han abierto oficinas en el Este .

81. Todo el mundo sabe que este verano los grandes corredores acudirán como un solo hombre al Tour_de_Francia y que , un mes después de haberse disputado , prácticamente nadie va a tener ganas de pensar en la Vuelta a España .

82. Toledo anunció hoy en rueda de prensa que no permitirá que Fujimori ejerza un nuevo mandato .

83. Pero sí creo que la vida de Moyano , su entereza hasta el final y su coraje , forma_parte del legado de los humanos , del inconsciente colectivo , de la sustancia común que todos somos .

84. Toledo señaló que el gobierno "quería llevarnos a una trampa en esta elección , querían repetir el mismo fraude de la primera vuelta" .

85. Y es que - una vez efectuado - se prenda fuego al conjunto , para , al fulgor de las llamas , celebrar un rito ancestral con la mente en_blanco .

86. Dice Antonio_Romero que para IU estas elecciones andaluzas son distintas a Las elecciones anteriores dentro y fuera_de esa comunidad autónoma .

87. Queda por examinar una forma de guerra biológica muy especial , bautizada con el nombre de guerra ecológica .

88. Además , el ideal de debilitar al_máximo nuestras inclinaciones sería lo más contrario a su afirmación de que Las disposiciones racionales del hombre quedarían dormidas para siempre - en su germen sin la tensión de fuerzas de la ambición , codicia , de la vanidad que rivaliza en la envidia , del apetito insaciable de posesión o de dominación .

89. Ahora será la viceministra Margarita_Robles quien tome las riendas de la negociación y se convierta en la interlocutora de María_Eugenia_Cuenca .

90. Ciertamente la demanda de créditos es enorme .

91. La bandera , que no falte .

92. La principal oposición al acuerdo se origina en los sectores sindicales que temen que su aplicación signifique la pérdida de miles de fuentes de trabajo en el país .

93. Los que gobiernan están torpes , sin reflejos , asustados .

94. Zülle , cabeza de la ONCE , capea el temporal .

95. Los datos espectroscópicos , y en algún caso el difractograma de rayos_X , indican que estas moléculas no son totalmente planas y se encuentran bastante deformadas .

96. Piden algo tan sencillo como la promoción de la mujer en el trabajo , ayudas para las inmigrantes y garantizar el cobro de las pensiones de las mujeres divorciadas .

97. Añadió que el Gobierno entiende que éste es un problema social , por lo que se aplican medidas para garantizar la seguridad y la integridad de los inmigrantes mediante los grupos especiales de protección denominados "grupos beta" .

98. Nos los imaginamos juntitos en las subidas , arañándose recíproca y alternativamente unos segundos en esos finales en los que se echa el resto "en persecución de Rominger" y acatando el juicio sumarísimo de la contrarreloj de Segovia .

99. De esa oposición extrema baste aquí una muestra .

100. De Latinoamérica figura el presidente de la Asamblea_Nacional_de_Nicaragua , Iván_Escobar .