Large-Scale Induction and Evaluation of Lexical Resources from the Penn-II and Penn-III Treebanks

Ruth O'Donovan*
Dublin City University

Michael Burke*†
Dublin City University

Aoife Cahill*
Dublin City University

Josef van Genabith*†
Dublin City University

Andy Way*†
Dublin City University

We present a methodology for extracting subcategorization frames based on an automatic lexical-functional grammar (LFG) f-structure annotation algorithm for the Penn-II and Penn-III Treebanks. We extract syntactic-function-based subcategorization frames (LFG semantic forms) and traditional CFG category-based subcategorization frames as well as mixed function/category-based frames, with or without preposition information for obliques and particle information for particle verbs. Our approach associates probabilities with frames conditional on the lemma, distinguishes between active and passive frames, and fully reflects the effects of long-distance dependencies in the source data structures. In contrast to many other approaches, ours does not predefine the subcategorization frame types extracted, learning them instead from the source data. Including particles and prepositions, we extract 21,005 lemma frame types for 4,362 verb lemmas, with a total of 577 frame types and an average of 4.8 frame types per verb. We present a large-scale evaluation of the complete set of forms extracted against the full COMLEX resource. To our knowledge, this is the largest and most complete evaluation of subcategorization frames acquired automatically for English.

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.

* National Centre for Language Technology, School of Computing, Dublin City University, Glasnevin, Dublin 9, Ireland. E-mail: {rodonovan,mburke,acahill,josef,away}@computing.dcu.ie.
† Centre for Advanced Studies, IBM, Dublin, Ireland.

Submission received: 19 March 2004; revised submission received: 18 December 2004; accepted for publication: 2 March 2005.

Extensive lexical resources, therefore, are crucial in the construction of wide-coverage computational systems based on such theories.

One important type of lexical information is the subcategorization requirements of an entry (i.e., the arguments a predicate must take in order to form a grammatical construction). Lexicons, including subcategorization details, were traditionally produced by hand. However, as the manual construction of lexical resources is time consuming, error prone, expensive, and rarely ever complete, it is often the case that the limitations of NLP systems based on lexicalized approaches are due to bottlenecks in the lexicon component. In addition, subcategorization requirements may vary across linguistic domain or genre (Carroll and Rooth 1998). Manning (1993) argues that, aside from missing domain-specific complementation trends, dictionaries produced by hand will tend to lag behind real language use because of their static nature. Given these facts, research on automating acquisition of dictionaries for lexically based NLP systems is a particularly important issue.

Aside from the extraction of theory-neutral subcategorization lexicons, there has also been work in the automatic construction of lexical resources which comply with the principles of particular linguistic theories such as LTAG, CCG, and HPSG (Chen and Vijay-Shanker 2000; Xia 1999; Hockenmaier, Bierner, and Baldridge 2004; Nakanishi, Miyao, and Tsujii 2004). In this article we present an approach to automating the process of lexical acquisition for LFG (i.e., grammatical-function-based systems). However, our approach also generalizes to CFG category-based approaches. In LFG, subcategorization requirements are enforced through semantic forms specifying which grammatical functions are required by a particular predicate. Our approach is based on earlier work on LFG semantic form extraction (van Genabith, Sadler, and Way 1999) and recent progress in automatically annotating the Penn-II and Penn-III Treebanks with LFG f-structures (Cahill et al. 2002; Cahill, McCarthy, et al. 2004). Our technique requires a treebank annotated with LFG functional schemata. In the early approach of van Genabith, Sadler, and Way (1999), this was provided by *manually* annotating the rules extracted from the publicly available subset of the AP Treebank to automatically produce corresponding f-structures. If the f-structures are of high quality, reliable LFG semantic forms can be generated quite simply by recursively reading off the subcategorizable grammatical functions for each local PRED value at each level of embedding in the f-structures. The work reported in van Genabith, Sadler, and Way (1999) was small scale (100 trees) and proof of concept and required considerable manual annotation work. It did not associate frames with probabilities, discriminate between frames for active and passive constructions, properly reflect the effects of long-distance dependencies (LDDs), or include CFG category information. In this article we show how the extraction process can be scaled to the complete *Wall Street Journal* (WSJ) section of the Penn-II Treebank, with about one million words in 50,000 sentences, based on the *automatic* LFG f-structure annotation algorithm described in Cahill et al. (2002) and Cahill, McCarthy, et al. (2004). More recently we have extended the extraction approach to the larger, domain-diverse Penn-III Treebank. Aside from the parsed WSJ section, this version of the treebank contains parses for a subsection of the Brown corpus (almost 385,000 words in 24,000 trees) taken from a variety of text genres.[1] In addition to extracting grammatical-function-

1 For the remainder of this work, when we refer to the *Penn-II Treebank*, we mean the parse-annotated WSJ, and when we refer to the *Penn-III Treebank*, we mean the parse-annotated WSJ and Brown corpus combined.

based subcategorization frames, we also include the syntactic categories of the predicate and its subcategorized arguments, as well as additional details such as the prepositions required by obliques and particles accompanying particle verbs. Our method discriminates between active and passive frames, properly reflects LDDs in the source data structures, assigns conditional probabilities to the semantic forms associated with each predicate, and does not predefine the subcategorization frames extracted.

In Section 2 of this article, we briefly outline LFG, presenting typical lexical entries and the encoding of subcategorization information. Section 3 reviews related work in the area of automatic subcategorization frame extraction. Our methodology and its implementation are presented in Section 4. In Section 5 we present results from the extraction process. We evaluate the complete induced lexicon against the COMLEX resource (Grishman, MacLeod, and Meyers 1994) and present the results in Section 6. To our knowledge, this is by far the largest and most complete evaluation of subcategorization frames automatically acquired for English. In Section 7, we examine the coverage of our lexicon in regard to unseen data and the rate at which new lexical entries are learned. Finally, in Section 8 we conclude and give suggestions for future work.

2. Subcategorization in LFG

Lexical functional grammar (Kaplan and Bresnan 1982; Bresnan 2001; Dalrymple 2001) is a member of the family of constraint-based grammars. It posits minimally two levels of syntactic representation:[2] c(onstituent)-structure encodes details of surface syntactic constituency, whereas f(unctional)-structure expresses abstract syntactic information about predicate–argument–modifier relations and certain morphosyntactic properties such as tense, aspect, and case. C-structure takes the form of phrase structure trees and is defined in terms of CFG rules and lexical entries. F-structure is produced from functional annotations on the nodes of the c-structure and implemented in terms of recursive feature structures (attribute–value matrices). This is exemplified by the analysis of the string *The inquiry soon focused on the judge* (wsj_0267_72) using the grammar in Figure 1, which results in the annotated c-structure and f-structure in Figure 2.

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 an **argument list** $\langle gf_1, gf_2, \ldots, gf_n\rangle$ specifying the **governable grammatical functions** (or arguments) required by the predicate to form a grammatical construction. In Figure 1 the verb *FOCUS* requires a subject and an oblique object introduced by the preposition *on*: $\text{FOCUS}\langle(\uparrow \text{SUBJ})(\uparrow \text{OBL}_{on})\rangle$. The argument list can be empty, as in the PRED value for *judge* in Figure 1. According to Dalrymple (2001), LFG assumes the following universally available inventory of grammatical functions: SUBJ(ect), OBJ(ect), OBJ_θ, COMP, XCOMP, OBL(ique)_θ, ADJ(unct), XADJ. OBJ_θ and OBL_θ represent families of grammatical functions indexed by their semantic role, represented by the theta subscript. This list of grammatical functions is divided into governable (subcategorizable) grammatical functions (**arguments**) and nongovernable (nonsubcategorizable) grammatical functions (**modifiers/adjuncts**), as summarized in Table 1.

2 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 1
Sample LFG rules and lexical entries.

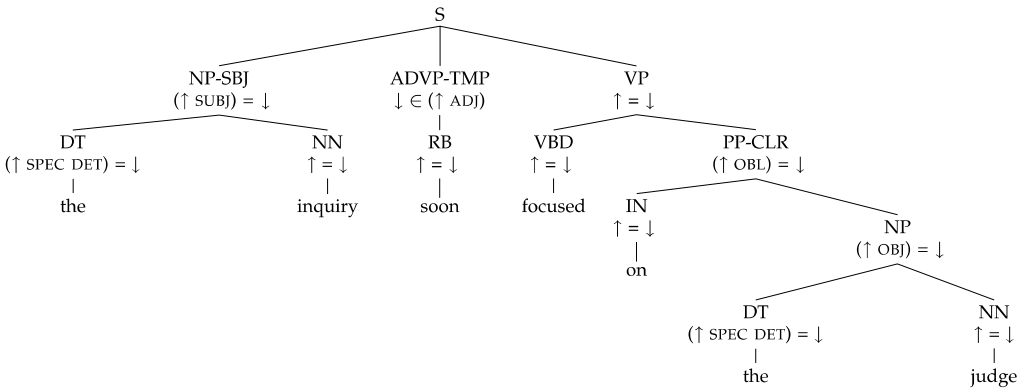
A number of languages allow the possibility of object functions in addition to the primary OBJ, such as the second or indirect object in English. Oblique arguments are realized as prepositional phrases in English. COMP, XCOMP, and XADJ are all clausal functions which differ in the way in which they are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

The judge thinks [COMP that it 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 [XCOMP to open an inquiry].

While many linguistic theories state subcategorization requirements in terms of phrase structure (CFG categories), Dalrymple (2001) questions the viability and universality of such an approach because of the variety of ways in which grammatical functions may be realized at the language-specific constituent structure level. LFG argues that subcategorization requirements are best stated at the f-structure level, in functional rather than phrasal terms. This is because of the assumption that abstract grammatical functions are primitive concepts as opposed to derivatives



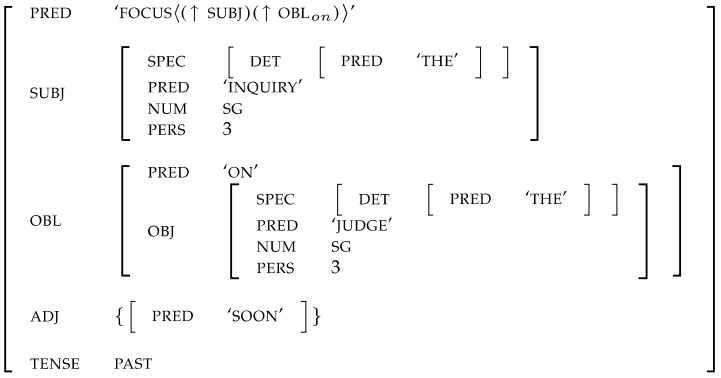


Figure 2
C- and f-structures for Penn Treebank sentence wsj_0267_72, *The inquiry soon focused on the judge*.

of phrase structural position. In LFG, the subcategorization requirements of a particular predicate are expressed by its semantic form: FOCUS⟨(↑ SUBJ)(↑ OBL$_{on}$)⟩ in Figure 1.

The subcategorization 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 contains all the governable grammatical functions that its predicate governs. An f-structure is *complete* iff it and all its subsidiary f-structures are locally complete. An f-structure is *locally coherent* iff all the governable grammatical functions that it contains are governed by a local predicate. An f-structure is *coherent* iff it and all its subsidiary f-structures are locally coherent. (page 211)

Consider again the f-structure in Figure 2. The semantic form associated with the verb `focus` is FOCUS⟨(↑ SUBJ)(↑ OBL$_{on}$)⟩. 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.

Table 1
Governable and nongovernable grammatical functions in LFG.

Governable GFs	Nongovernable GFs
SUBJ	ADJ
OBJ	XADJ
XCOMP	
COMP	
OBJ_θ	
OBL_θ	

Because of the specific form of the LFG lexicon, our extraction approach differs in interesting ways from that of previous lexical extraction experiments. This contrast is made evident in Sections 3 and 4.

3. Related Work

The encoding of verb subcategorization 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 and 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 lexicons automatically, especially for English. In this section, we discuss approaches to CFG-based subcategorization 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 evaluation of these approaches is discussed in greater detail in Section 6, in which we compare our results with those reported elsewhere in the literature.

We will divide more-general approaches to subcategorization frame acquisition into two groups: those which extract information from raw text and those which use preparsed and hand-corrected treebank data as their input. Typically in the approaches based on raw text, a number of subcategorization patterns are predefined, a set of verb subcategorization frame associations are hypothesized from the data, and statistical methods are applied to reliably select hypotheses for the final lexicon. Brent (1993) relies on morphosyntactic cues in the untagged Brown corpus as indicators of six predefined subcategorization frames. The frames do not include details of specific prepositions. Brent used hypothesis testing on binomial frequency data to statistically filter the induced frames. Ushioda et al. (1993) run a finite-state NP parser on a POS-tagged corpus to calculate the relative frequency of the same six subcategorization 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 subcategorization frame recognition) in order to extract verbs and the constituents with which they co-occur. He assumes 19 different subcategorization

frame definitions, and the extracted frames include details of specific prepositions. The extracted frames are noisy as a result of parser errors and so are filtered using the binomial hypothesis theory (BHT), following Brent (1993). Applying his technique to approximately four million words of *New York Times* newswire, Manning acquired 4,900 verb-subcategorization frame pairs for 3,104 verbs, an average of 1.6 frames per verb. Briscoe and Carroll (1997) predefine 163 verbal subcategorization frames, obtained by manually merging the classes exemplified in the COMLEX (MacLeod, Grishman, and Meyers 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. Briscoe and Carroll (1997) refine the BHT with a priori information about the probabilities of subcategorization 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. Carroll and Rooth (1998) use a handwritten head-lexicalized, context-free grammar and a text corpus to compute the probability of particular subcategorization patterns. The approach is iterative with the aim of estimating the distribution of subcategorization 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.

More recently, a number of researchers have applied similar techniques to automatically derive lexical resources for languages other than English. Schulte im Walde (2002a, 2002b) uses a head-lexicalized probabilistic context-free grammar similar to that of Caroll and Rooth (1998) to extract subcategorization frames from a large German newspaper corpus from the 1990s. 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 nonfinite 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. Because of computing time constraints, Schulte im Walde limits sentence length for grammar training and parsing. Sentences of length between 5 and 10 words were used to bootstrap the lexicalized grammar model. For lexicalized training, sentences of length between 5 and 13 words were used. The result is a subcategorization lexicon for over 14,000 German verbs. The extensive evaluation carried out by Schulte im Walde will be discussed in greater detail in Section 6.

Approaches using treebank-based data as a source for subcategorization information, such as ours, 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 more than 150 different sequences of syntactic and functional tags in the treebank. Each of these sequences was categorized 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. Sarkar and Zeman (2000) present an approach to learn previously unknown frames for Czech from the Prague Dependency Bank (Hajic

1998). Czech is a language with a freer word order 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, whereas a subcategorization frame contains a subset of the dependents in the observed frame. Finding subcategorization 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 subcategorization frames from 19,126 sentences for 914 verbs (those which occurred five times or more). Marinov and Hemming (2004) present preliminary work on the automatic extraction of subcategorization frames for Bulgarian from the BulTreeBank (Simov, Popova, and Osenova 2002). In a similar way to that of Sarkar and Zeman (2000), Marinov and Hemming's system collects both arguments and adjuncts. It then uses the binomial log-likelihood ratio 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 and Hemming are working with a preliminary version of the treebank with 580 sentences.

Work has been carried out on the extraction of formalism-specific lexical resources from the Penn-II Treebank, in particular TAG, CCG, and HPSG. As these formalisms are fully lexicalized 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 lexicalized TAG from the Penn-II Treebank with the aim of constructing a statistical model for parsing. The extraction procedure utilizes a head percolation table as introduced by Magerman (1995) in combination with a variation of Collins's (1997) approach 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 2,366 to 8,996. Xia (1999) also presents a similar method for the extraction of a TAG from the Penn Treebank. The extraction procedure consists of three steps: First, the bracketing of the trees in the Penn Treebank is corrected and extended based on the approaches of Magerman (1994) and Collins (1997). 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 3,014 to 6,099.

Hockenmaier, Bierner, and Baldridge (2004) outline a method for the automatic extraction of a large syntactic CCG lexicon from the Penn-II Treebank. For each tree, the algorithm annotates the nodes 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 (1997). Each node is subsequently assigned the relevant 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 LDDs. Unlike our approach, those of Xia (1999) and Hockenmaier, Bierner, and Baldridge (2004) include a substantial initial correction and clean-up of the Penn-II trees.

Miyao, Ninomiya, and Tsujii (2004) and Nakanishi, Miyao, and Tsujii (2004) describe a methodology for acquiring an English HPSG from the Penn-II Treebank. Manually defined heuristics are used to automatically annotate each tree in the treebank with partially specified HPSG derivation trees: Head/argument/modifier distinctions are made for each node in the tree based on Magerman (1994) and Collins (1997);

the whole tree is then converted to a binary tree; heuristics are applied to deal with phenomena such as LDDs and coordination and to correct some errors in the treebank, and finally an HPSG category is assigned to each node in the tree in accordance with its CFG category. In the next phase of the process (externalization), HPSG lexical entries are automatically extracted from the annotated trees through the application of "inverse schemata."

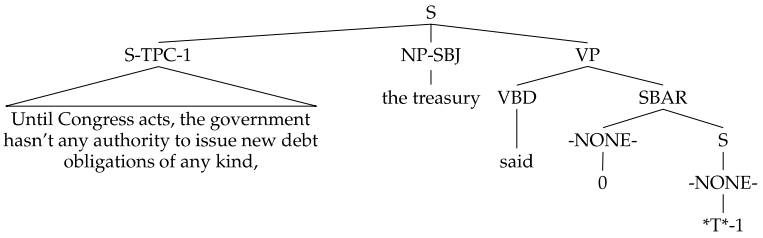
4. Methodology

The first step in the application of our methodology is the production of a treebank annotated with LFG f-structure information. F-structures are attribute–value structures which represent abstract syntactic information, approximating to basic predicate–argument–modifier structures. Most of the early work on automatic f-structure annotation (e.g., van Genabith, Way, and Sadler 1999; Frank 2000; Sadler, van Genabith, and Way 2000) was applied only to small data sets (fewer than 200 sentences) and was largely proof of concept. However, more recent work (Cahill et al. 2002; Cahill, McCarthy, et al. 2004) has presented efforts in evolving and scaling up annotation techniques to the Penn-II Treebank (Marcus et al. 1994), containing more than 1,000,000 words and 49,000 sentences.

We utilize the automatic annotation algorithm of Cahill et al. (2002) and Cahill, McCarthy, et al. (2004) to derive a version of Penn-II in which each node in each tree is annotated with LFG functional annotations in the form of attribute-value structure equations. The algorithm uses categorial, configurational, local head, and Penn-II functional and trace information. The annotation procedure is dependent on locating the head daughter, for which an amended version of Magerman (1994) is used. The head is annotated with the LFG equation $\uparrow=\downarrow$. Linguistic generalizations are provided over the left (the prefix) and the right (suffix) context of the head for each syntactic category occurring as the mother nodes of such heads. To give a simple example, the rightmost NP to the left of a VP head under an S is likely to be the subject of the sentence ($\uparrow$ SUBJ $=\downarrow$), while the leftmost NP to the right of the V head of a VP is most probably the verb's object ($\uparrow$ OBJ $=\downarrow$). Cahill, McCarthy, et al. (2004) provide four classes of annotation principles: one for noncoordinate configurations, one for coordinate configurations, one for traces (long-distance dependencies), and a final "catch all and clean up" phase.

The satisfactory treatment of long-distance dependencies by the annotation algorithm is imperative for the extraction of accurate semantic forms. The Penn Treebank employs a rich arsenal of traces and empty productions (nodes which do not realize any lexical material) to coindex displaced material with the position where it should be interpreted semantically. The algorithm of Cahill, McCarthy, et al. (2004) translates the traces into corresponding reentrancies in the f-structure representation by treating null constituents as full nodes and recording the traces in terms of `index=i` f-structure annotations (Figure 3). Passive movement is captured and expressed at f-structure level using a `passive:+` annotation. Once a treebank tree is annotated with feature structure equations by the annotation algorithm, the equations are collected, and a constraint solver produces an f-structure.

In order to ensure the quality of the semantic forms extracted by our method, we 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 2. Both use the evaluation software and triple 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



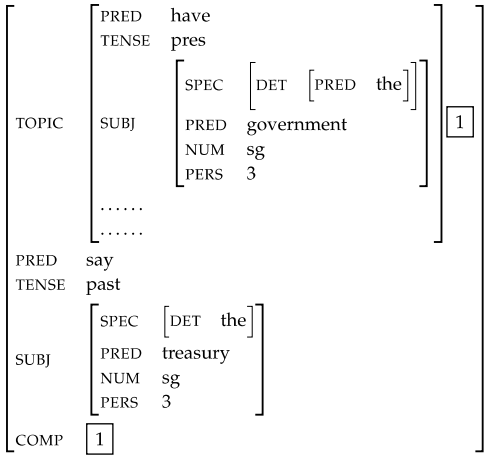


Figure 3
Use of reentrancy between TOPIC and COMP to capture long-distance *dependency* in Penn Treebank sentence wsj_0008_2, *Until Congress acts, the government hasn't any authority to issue new debt obligations of any kind, the Treasury said.*

from Section 23 of the Penn Treebank as described in Cahill, McCarthy, et al. (2004). For the full set of annotations they 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, Cahill, 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. They report precision of over 88.5% and recall of over 86% (Table 2). The PARC 700 Dependency Bank differs substantially from both the DCU 105 f-structure bank and the automatically generated f-structures in regard to

Table 2
Results of f-structure evaluation.

	DCU 105	PARC 700
Precision	96.52%	88.57%
Recall	96.62%	86.10%
F-score	96.57%	87.32%

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 in this process and a full analysis of results is presented in Burke, Cahill, et al. (2004a). Results broken down by grammatical function for the DCU 105 evaluation are presented in Table 3. OBL (prepositional phrase) arguments are traditionally difficult to annotate reliably. The results show, however, that with respect to obliques, the annotation algorithm, while slightly conservative (recall of 82%), is very accurate: 96% of the time it annotates an oblique, the annotation is correct.

A high-quality set of f-structures having been produced, the semantic form extraction methodology is applied. This is based on and substantially extends both the granularity and coverage of an idea in van Genabith, Sadler, and Way (1999):

> For each f-structure generated, for each level of embedding we determine the local PRED value and collect the subcategorisable grammatical functions present at that level of embedding. (page 72)

Consider the automatically generated f-structure in Figure 4 for tree wsj_0003_22 in the Penn-II and Penn-III Treebanks. It is crucial to note that in the automatically generated f-structures the value of the PRED feature is a lemma and not a semantic form. Exploiting the information contained in the f-structure and applying the method described above, we recursively extract the following nonempty semantic forms: `impose([subj, obj, obl:on])`, `in([obj])`, `of([obj])`, and `on([obj])`. In effect, in both the approach of van Genabith, Sadler, and Way (1999) and our approach, semantic forms are reverse-engineered from automatically generated f-structures for treebank trees. The automatically induced semantic forms contain the following subcategorizable syntactic functions:

SUBJ	OBJ	OBJ2	OBL_{prep}	OBL2	COMP	XCOMP	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

Table 3
Precision and recall on automatically generated f-structures by feature against the DCU 105.

Feature	Precision	Recall	*F*-score
ADJUNCT	892/968 = 92	892/950 = 94	93
COMP	88/92 = 96	88/102 = 86	91
COORD	153/184 = 83	153/167 = 92	87
DET	265/267 = 99	265/269 = 99	99
OBJ	442/459 = 96	442/461 = 96	96
OBL	50/52 = 96	50/61 = 82	88
OBLAG	12/12 = 100	12/12 = 100	100
PASSIVE	76/79 = 96	76/80 = 95	96
RELMOD	46/48 = 96	46/50 = 92	94
SUBJ	396/412 = 96	396/414 = 96	96
TOPIC	13/13 = 100	13/13 = 100	100
TOPICREL	46/49 = 94	46/52 = 88	91
XCOMP	145/153 = 95	145/146 = 99	97

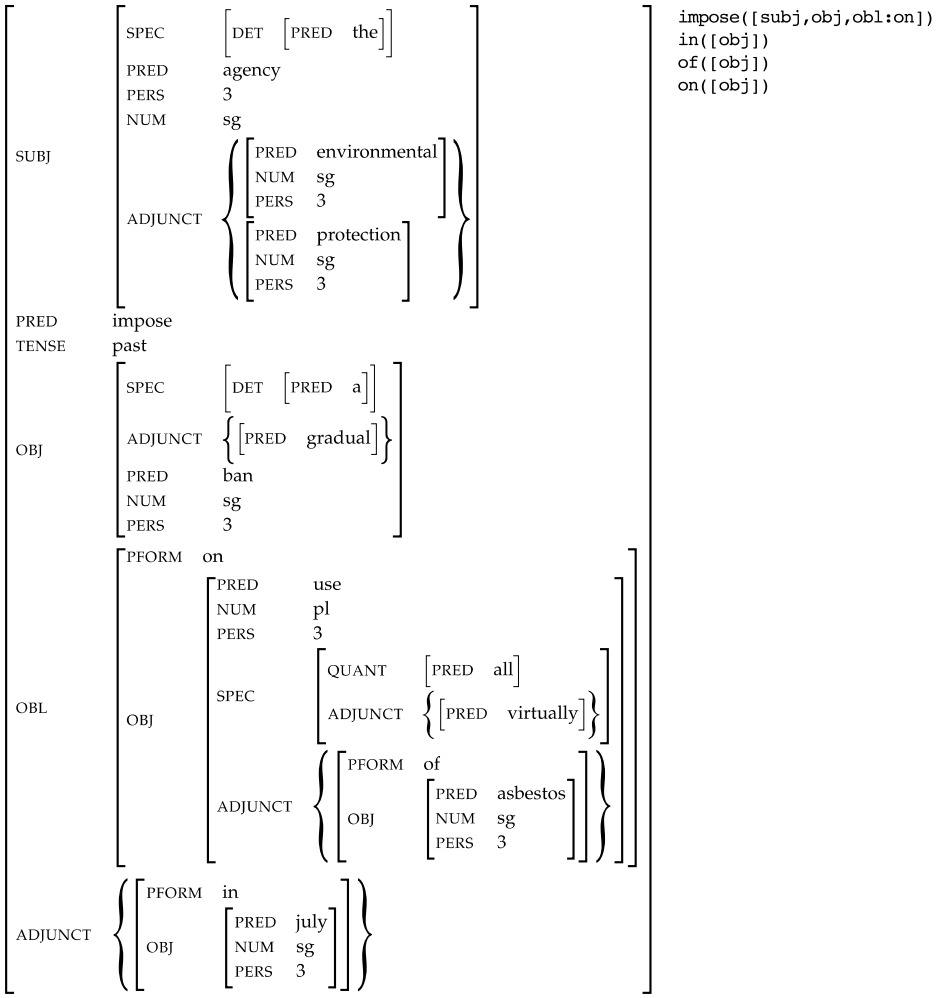


Figure 4
Automatically generated f-structure and extracted semantic forms for the Penn-II Treebank string wsj_0003_22, *In July, the Environmental Protection Agency imposed a gradual ban on virtually all uses of asbestos.*

OBL$_{prep}$ includes the prepositional head of the PP, PART includes the actual particle which occurs, for example, `add([subj,obj,part:up])`.

In the work presented here, we substantially extend and scale the approach of van Genabith, Sadler, and Way (1999) in regard to coverage, granularity, and evaluation. First, we scale the approach to the full WSJ section of the Penn-II Treebank and the parsed Brown corpus section of Penn-III, with a combined total of approximately 75,000 trees. Van Genabith, Sadler, and Way (1999) was proof of concept on 100 trees. Second, in contrast to the approach of van Genabith, Sadler, and Way (1999) (and many other approaches), our approach fully reflects long-distance dependencies, indicated in terms of traces in the Penn-II and Penn-III Treebanks and corresponding reentrancies at f-structure. Third, in addition to abstract syntactic-function-based subcategorization frames, we also compute frames for syntactic function–CFG category pairs, for both the verbal heads and their arguments, and also generate

Table 4
Conflation of Penn Treebank tags.

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

pure CFG-based subcategorization frames. Fourth, in contrast to the approach of van Genabith, Sadler, and Way (1999) (and many other approaches), our method differentiates between frames for active and passive constructions. Fifth, in contrast to that of van Genabith, Sadler, and Way (1999), our method associates conditional probabilities with frames. Sixth, we evaluate the complete set of semantic forms extracted (not just a selection) against the manually constructed COMLEX (MacLeod, Grishman, and Meyers 1994) resource.

In order to capture CFG-based categorial information, we add a CAT feature to the f-structures automatically generated from the Penn-II and Penn-III Treebanks. 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 reduces the risk of inaccurate assignment of subcategorization frame frequencies due to POS ambiguity, distinguishing, for example, between the nominal and verbal occurrences of the lemma `fight`. With this, the output for the verb `impose` in Figure 4 is `impose(`**`v`**`,[subj, obj, obl:on])`. For some of our experiments, we conflate the different verbal (and other) tags used in the Penn Treebanks to a single verbal marker (Table 4). As a further extension, the extraction procedure reads off the syntactic category of the head of each of the subcategorized syntactic functions: `impose(`**`v`**`,[subj(`**`n`**`),obj(`**`n`**`),obl:on])`.[3] In this way, our methodology is able to produce surface syntactic as well as abstract functional subcategorization details. Dalrymple (2001) argues that there are cases, albeit exceptional ones, in which constraints on syntactic category are an issue in subcategorization. In contrast to much of the work reviewed in Section 3, which limits itself to the extraction of surface syntactic subcategorization details, our system can provide this information as well as details of grammatical function.

3 We do not associate syntactic categories with OBLs as they are always PPs.

Another way in which we develop and extend the basic extraction algorithm is to deal with passive voice and its effect on subcategorization behavior. Consider Figure 5: 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)*. The annotation algorithm's accuracy in recognizing passive constructions is reflected by the f-score of 96% reported in Table 3 for the PASSIVE feature.

The syntactic functions COMP and XCOMP refer to clausal complements with different predicate control patterns as described in Section 2. However, as it stands, neither of these functions betrays 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 subcategorized clausal arguments, differentiating, for example, between a *that*-clause and a *to* + *infinitive* clause (Ushioda et al. 1993). With only a slight modification, our system, along with the details provided by the automatically generated f-structures, allows us to extract frames with an equivalent level of detail. For example, to identify a *that*-clause, we use

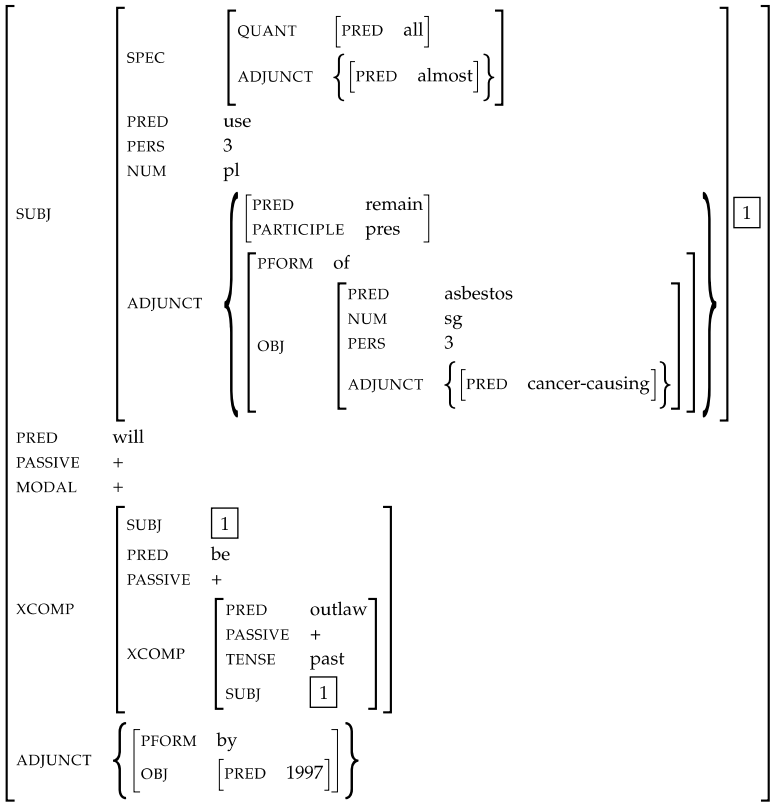


Figure 5
Automatically generated f-structure for the Penn-II Treebank string wsj_0003_23. *By 1997, almost all remaining uses of cancer-causing asbestos will be outlawed.*

Table 5
Semantic forms for the verb `accept`.

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

the feature-value pair `that:+` at f-structure level to read off the following subcategorization frame for the verb *add: add([subj,comp(that)])*. Using the feature-value pair *to_inf:+*, we can identify *to + infinitive* clauses, resulting in the following frame for the verb *want: want([subj,xcomp(to_inf)])*. We can also derive control information about open complements. In Figure 5, the reentrant XCOMP subject is identical to the subject of `will` in the matrix clause, which allows us to induce information about the nature of the external control of the XCOMP (i.e., whether it is subject or object control).

In order to estimate the likelihood of the co-occurrence of a predicate with a particular argument list, we compute conditional probabilities for subcategorization 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. Because of variations in verbal subcategorization across domains, probabilities are also useful for predicting the way in which verbs behave in certain contexts. In Section 6, we use the conditional probabilities to filter possible error judgments by our system. Tables 5–7 show, with varying levels of analysis, the attested semantic forms for the verb `accept` with their associated conditional probabilities. 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* (shown in boldface type) in Tables 5 and 6.[4] Table 7 shows the joint grammatical-function/syntactic-category-based subcategorization frames.

5. Results

We extract semantic forms for 4,362 verb lemmas from Penn-III. Table 8 shows the number of distinct semantic form types (i.e., lemma and argument list combination)

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

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

Table 6
Semantic forms for the verb `accept` marked with `p` for passive use.

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 7
Semantic forms for the verb `accept` including syntactic category for each grammatical function.

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

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 parameterized for prepositions and particles are included for particle verbs, we 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. Unlike many of the researchers whose work is reviewed in Section 3, we do not predefine the frames extracted by our system. Table 9 shows the numbers of distinct frame types extracted from Penn-II, ignoring PRED values.[5] We provide two columns of statistics, one in which all oblique (PP) arguments are condensed into one OBL function and all particle arguments are condensed into `part`, and the other in which we differentiate among `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, we extract 38 frame types. Discriminating particles and obliques by preposition, we extract 577 frame types. Table 10 shows the same results for Penn-III, with 50 simple frame types and 1,084 types when parameterized for prepositions and particles. We also show the result of applying absolute thresholding techniques to the semantic forms induced. Applying an absolute threshold of five occurrences, we still generate 162 frame types

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

Table 8
Number of semantic form types for Penn-III.

	Without prepositions and particles	With prepositions and particles
Semantic form types	15,166	21,005
Active	11,038	16,000
Passive	4,128	5,005

Table 9
Number of frame types for verbs for Penn-II.

	Without prepositions and particles	With prepositions and particles
Number of frame types	38	577
Number of singletons	1	243
Number occurring twice	1	84
Number occurring five or fewer times	7	415
Number occurring more than five times	31	162

from Penn-II and 221 from Penn-III. Briscoe and Carroll (1997), by comparison, employ 163 distinct predefined frames.

6. Evaluation

Most of the previous approaches discussed in Section 3 have been evaluated to different degrees. In general, a small number of frequently occurring verbs is selected, and the subcategorization 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, Grishman, and Meyers 1994). There are advantages and disadvantages to both types of gold standard. While it is time-consuming to manually construct a custom-made standard, the resulting standard has the advantage of containing only the subcategorization frames exhibited in the test data. Using an existing externally produced resource is quicker, but the gold

Table 10
Number of frame types for verbs for Penn-III.

	Without prepositions and particles	With prepositions and particles
Number of frame types	50	1,084
Number of singletons	6	544
Number occurring twice	2	147
Number occurring five or fewer times	12	863
Number occurring more than five times	38	221

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 recognizes 15 frames, and these do not contain details of subcategorized-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 Section 3. Sarkar 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 subcategorization 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 3,000 German verbs with a token frequency between 10 and 2,000 against the Duden (Dudenredaktion 2001). We will refer to this work and the methods and results presented by Schulte im Walde again in Sections 6.2 and 6.3.

We carried out a large-scale evaluation of our automatically induced lexicon (2,993 active verb lemmas for Penn-II and 3,529 for Penn-III, as well as 1,422 passive verb lemmas from Penn-II) against the COMLEX resource. To our knowledge this is the most extensive evaluation ever carried out for English lexical extraction. We conducted a number of experiments on the subcategorization frames extracted from Penn-II and Penn-III which are described and discussed in Sections 6.2, 6.3, and 6.4. Finding a common format for the gold standard and induced lexical entries is a nontrivial task. To ensure that we did not bias the evaluation in favor of either resource, we 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 we carried out six basic experiments (and two additional ones for COMLEX-LFG Mapping II) for the active subcategorization 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 we also evaluated the automatically extracted passive frames and experimented with absolute thresholds. Direct comparison of subcategorization frame acquisition systems is difficult because of 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 subcategorization frames (e.g., whether they are parameterized for specific prepositions). Therefore, in order to establish a baseline against which to compare our results, following Schulte in Walde (2002b), we 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.

6.1 COMLEX

We evaluate our induced semantic forms against COMLEX (MacLeod, Grishman, and Meyers 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 subcategorization information than is available in either the OALD or the LDOCE (Proctor 1978), both for verbs and for nouns

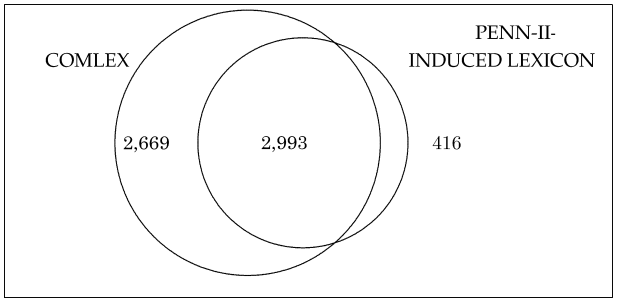


Figure 6
Intersection between active-verb lemma types in COMLEX and the Penn-II-induced lexicon.

and adjectives which take complements (Grishman, MacLeod, and Meyers 1994). By choosing to evaluate against COMLEX, we set our sights high: Our extracted semantic forms are fine-grained, and COMLEX is considerably more detailed than the OALD or LDOCE used for earlier evaluations. While our system can generate semantic forms for any lemma (regardless of part of speech) which induces a PRED value, we have thus far evaluated the automatic generation of subcategorization 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 subcategorization details, it was necessary for us to extract the subcategorization 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 organized 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 behavior 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 subcategorization 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). (Note that the details of the subject are not included in COMLEX frames.) 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 like the following:

```
(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 subcategorization frame, which lists the syntactic CF-PSG constituents in sequence (omitting the subject, 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

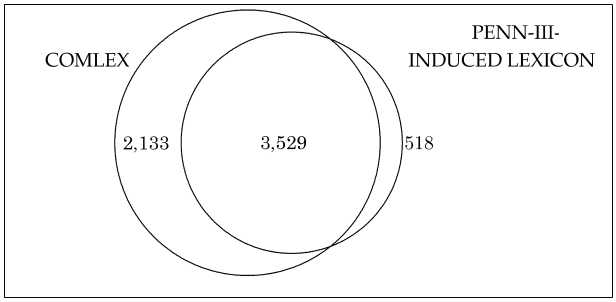


Figure 7
Intersection between active-verb lemma types in COMLEX and the Penn-III-induced lexicon.

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 we induce.

We present the evaluation for the verbs which occur in an active context in the treebank. COMLEX does not provide passive frames. For Penn-II, there are 2,993 verb lemmas (used actively) that both resources have in common. 2,669 verb lemmas appear in COMLEX but not in the induced lexicon, and 416 verb lemmas (used actively) appear in the induced lexicon but not in COMLEX (Figure 6). For Penn-III, COMLEX and the induced lexicon share 3,529 verb lemmas (used actively). This is shown in Figure 7. [6]

6.2 COMLEX-LFG Mapping I and Penn-II

In order to carry out the evaluation, we have to find a common format for the expression of subcategorization information between our induced LFG-style subcategorization 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 we 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 6.3), so we conflate our two LFG categories to that of COMP. The process is summarized in Table 11.

The manually constructed COMLEX entries provide a gold standard against which we evaluate the automatically induced frames. We calculate the number of true positives (*tp*s) (where our semantic forms and those from COMLEX are the same), the number of false negatives (*fn*s) (those frames which appeared in COMLEX but were not produced by our system), and the number of false positives (*fp*s) (those frames

6 Given these figures, one might 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: 518 lemmas "discovered" by the induction experiment are not listed in COMLEX; see the error analysis in Section 6.5.

Table 11
Mapping I: Merging of COMLEX and LFG syntactic functions.

Our syntactic functions	COMLEX syntactic functions	Merged function
SUBJ	Subject	SUBJ
OBJ	Object	OBJ
OBJ2	Obj2	$\texttt{OBJ}_i$
OBL	Obj3	
OBL2	Obj4	
COMP	Comp	COMP
XCOMP		
PART	Part	PART

produced by our system which do not appear in COMLEX). We calculate precision, recall, and F-score using the following standard equations:

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

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

$$f\text{-}score = \frac{2 \times recall \times precision}{recall + precision}$$

We use the frequencies associated with each of our semantic forms in order to set a **relative threshold** to filter the selection of semantic forms. For a threshold of 1% we disregard any semantic forms with a conditional probability (i.e., given a lemma) of less than or equal to 0.01. As some verbs occur less frequently than others, we think it is important to use a relative rather than absolute threshold (as in Carroll and Rooth [1998], for instance) in this way. We carried out the evaluation in a similar way to Schulte im Walde's (2002b) for German, the only experiment comparable in scale to ours. Despite the obvious differences in approach and language, this allows us to make some tentative comparisons between our respective results. The statistics shown in Table 12 give the results of three different experiments with the relative threshold set to 1%. As for all the results tables, the baseline statistics (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, and the results achieved by our induced lexicon are presented in the right column. Distinguishing between complement and adjunct prepositional phrases is a notoriously difficult aspect of automatic subcategorization frame acquisition. For this reason, following the evaluation setup in Schulte im Walde (2002b), the three experiments vary with respect to the amount of prepositional information contained in the subcategorization frames.

Experiment 1. Here we excluded subcategorized prepositional-phrase arguments entirely from the comparison. In a manner similar to that of Schulte im Walde (2002b), any

Table 12
Results of Penn-II evaluation of active frames against COMLEX (relative threshold of 1%).

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

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 12), our results (precision of 75.2%, recall of 69.1%, and F-score of 72.0%) are remarkably similar to those of Schulte im Walde (2002b), who reports precision of 74.53%, recall of 69.74%, and an f-score of 72.05%.

Experiment 2. Here we include subcategorized prepositional phrase arguments but only in their simplest form; that is, they were not parameterized for particular prepositions. For example, the frame `[subj,obl:for]` is rewritten as `[subj,obl]`. Using a relative threshold of 1% (Table 12), our results (precision of 65.5%, recall of 63.1%, and F-score of 64.3%) compare favorably to those of Schulte im Walde (2002b), who recorded precision of 60.76%, recall of 63.91%, and an F-score of 62.30%.

Experiment 3. Here we used semantic forms which contain details of specific prepositions for any subcategorized prepositional phrase (e.g., `[subj,obl:for]`). Using a relative threshold of 1% (Table 12), our precision figure (71.8%) is quite high (in comparison to 65.52% as recorded by Schulte im Walde [2002b]). However our recall (16.8%) is very low (compared to the 50.83% that Schulte im Walde [2002b] reports). Consequently our F-score (27.3%) is also low (Schulte im Walde [2002b] records an F-score of 57.24%). The reason for this is discussed in Section 6.2.1.

The statistics in Table 13 are the result of the second experiment, in which the relative 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 greater threshold means that there are fewer semantic forms associated with each verb in the induced lexicon, but they are more likely to be correct because of their greater frequency of occurrence. The conditional probabilities we associate with each semantic form together with thresholding can be used to customize the induced lexicon to the task for which it is required, that is, whether a very precise lexicon is preferred to one with broader

Table 13
Results of Penn-II evaluation of active frames against COMLEX (relative threshold of 5%).

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

coverage. In Tables 12 and 13, the baseline is exceeded in all experiments with the exception of Experiment 2. This can be attributed to Mapping I, in which OBL_i becomes OBJ_i (Table 11). Experiment 2 includes obliques without the specific preposition, meaning that in this mapping, 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 in which this effect disappears is presented in Section 6.3.

6.2.1 Directional Prepositions. Our recall statistic was particularly low in the case of evaluation using details of prepositions (Experiment 3, Tables 12 and 13). This can be accounted for by the fact that the creators of COMLEX have chosen to err on the side of overgeneration in regard to the list of prepositions which may occur with a verb and a subcategorization frame containing a prepositional phrase. This is particularly true of directional prepositions. For COMLEX, a list of 31 directional prepositions (Table 14) 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, MacLeod, and Meyers (1994) acknowledge 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, we incorporated this list of directional prepositions by default into our semantic form induction process in the same way as the creators of COMLEX have done. Table 15 shows that doing so results in a significant improvement in the recall statistic (45.1%), as would be expected, with the new statistic being almost three times as good as the result reported in Table 12 for Experiment 3 (16.8%). There is also an improvement in the precision figure (from 71.8% to 86.9%). This is due to a substantial increase in the number of true positives (from 5,612 to 14,675) compared with a stationary false positive figure (2,205 in both cases). The f-score increases from 27.3% to 59.4%.

6.3 COMLEX-LFG Mapping II and Penn-II

The COMLEX-LFG Mapping I presented above establishes a "least common denominator" for the COMLEX and our 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 subcategorization frames, we conducted a further set of experiments in which we converted the information in the COMLEX entries to the format of our extracted semantic forms. We explicitly differentiated between OBLs and OBJs by automatically

Table 14
COMLEX directional prepositions.

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	toward
up	up to	via	

Table 15
Penn-II evaluation of active frames against COMLEX using p-dir list (relative threshold of 1%).

Mapping I	Precision	Recall	*F*-score
Experiment 3	86.9%	45.1%	59.4%

deducing whether a COMLEX OBJ$_i$ was coindexed 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 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 summarized in Table 16. The results of the subsequent evaluation are presented in Tables 17 and 18. We 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 17 and 18 are slightly lower than those in Tables 12 and 13, changing the mapping in this way results in an increase in precision in each case (by as much as 11.6%). The results of the lexical evaluation are consistently better than the baseline, in some cases by almost 16% (Experiment 2, threshold 5%). Notice that in contrast to Tables 12 and 13, in the more-fine-grained COMLEX-LFG Mapping II presented here, all experiments exceed the baseline.

6.3.1 Directional Prepositions. The recall figures for Experiments 3 and 3a in Table 17 (24.0% and 21.5%) and Table 18 (19.7% and 17.4%) drop in a similar fashion to the results seen in Tables 12 and 13. For this reason, we again incorporated the list of 31 directional prepositions (Table 14) by default and reran Experiments 3 and 3a for a threshold of 1%. The results are presented in Table 19. The effect was as expected: The recall scores for the two experiments increased to 40.8% and 35.4% (from 24.0% and 22.5%), and the F-scores increased to 54.4% and 49.7% (from 35.9% and 33.0%).

6.3.2 Passive Evaluation. Table 20 presents the results of evaluating the extracted passive semantic forms for 1,422 verb lemmas shared by the induced lexicon and COMLEX.

Table 16
Mapping II: Merging of COMLEX and LFG syntactic functions.

Our syntactic functions	COMLEX syntactic functions	Merged function
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 17
Results of Penn-II evaluation of active frames against COMLEX (relative threshold of 1%).

	Precision		Recall		*F*-score	
Mapping II	Baseline	Induced	Baseline	Induced	Baseline	Induced
Experiment 1	72.1%	79.0%	58.5%	59.6%	64.6%	68.0%
Experiment 2	65.2%	77.1%	37.4%	50.4%	47.5%	61.0%
Experiment 2a	65.2%	76.4%	32.7%	44.5%	43.6%	56.3%
Experiment 3	65.2%	75.9%	15.2%	24.0%	24.7%	35.9%
Experiment 3a	65.2%	71.0%	13.6%	21.5%	22.5%	33.0%

Table 18
Results of Penn-II evaluation of active frames against COMLEX (relative threshold of 5%).

	Precision		Recall		*F*-score	
Mapping II	Baseline	Induced	Baseline	Induced	Baseline	Induced
Experiment 1	72.1%	83.5%	58.5%	54.7%	64.6%	66.1%
Experiment 2	65.2%	81.4%	37.4%	44.8%	47.5%	57.8%
Experiment 2a	65.2%	80.9%	32.7%	39.0%	43.6%	52.6%
Experiment 3	65.2%	75.9%	15.2%	19.7%	24.7%	31.3%
Experiment 3a	65.2%	75.5%	13.6%	17.4%	22.5%	28.3%

We applied lexical-redundancy rules (Kaplan and Bresnan 1982) to automatically convert the active COMLEX frames to their passive counterparts: For example, subjects are demoted to optional *by* oblique agents, and direct objects become subjects. The resulting precision was very high (from 72.3% to 80.2%), and there was the expected drop in recall when prepositional details were included (from 54.7% to 29.3%).

Table 19
Penn-II evaluation of active frames against COMLEX using p-dir list (relative threshold of 1%).

Mapping II	Precision	Recall	*F*-score
Experiment 3	81.7%	40.8%	54.4%
Experiment 3a	83.1%	35.4%	49.7%

Table 20
Results of Penn-II evaluation of passive frames (relative threshold of 1%).

Passive	Precision	Recall	*F*-score
Experiment 2	80.2%	54.7%	65.1%
Experiment 2a	79.7%	46.2%	58.5%
Experiment 3	72.6%	33.4%	45.8%
Experiment 3a	72.3%	29.3%	41.7%

6.3.3 Absolute Thresholds. Many of the previous approaches discussed in Section 3 use a limited number of verbs for evaluation, based on the verbs' absolute frequency in the corpus. We carried out a similar experiment. Table 21 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. The use of an absolute threshold results in an increase in precision (from 77.1% to 82.3% and 81.7%), an increase in recall (from 50.4% to 60.8% to 58.7%), and an overall increase in F-score (from 61.0% to 69.9% and 68.4%).

6.4 Penn-III (Mapping-II)

Recently we have applied our methodology to the Penn-III Treebank, a more balanced corpus resource with a number of text genres. Penn-III consists of the WSJ section from Penn-II as well as a parse-annotated subset of the Brown corpus. The Brown corpus comprises 24,242 trees compiled from a variety of text genres including popular lore, general fiction, science fiction, mystery and detective fiction, and humor. It has been shown (Roland and Jurafsky 1998) that the subcategorization tendencies of verbs vary across linguistic domains. Our aim, therefore, is to increase the scope of the induced lexicon not only in terms of the verb lemmas for which there are entries, but also in terms of the frames with which they co-occur. The f-structure annotation algorithm was extended with only minor amendments to cover the parsed Brown corpus. The most important of these was the way in which we distinguish between oblique and adjunct. We noted in Section 4 that our method of assigning an oblique annotation in Penn-II was precise, albeit conservative. Because of a change of annotation policy in Penn-III, the -CLR tag (indicating a close relationship between a PP and the local syntactic head), information which we had previously exploited, is no longer used. For Penn-III the algorithm annotates all PPs which do not carry a Penn adverbial functional tag (such as -TMP or -LOC) and 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.

When evaluating the application of the lexical extraction system on Penn-III, we carried out two sets of experiments, identical in each case to those described for Penn-II in Section 6.3, including the use of relative (1% and 5%) rather than absolute thresholds. For the first set of experiments we evaluated the lexicon induced from the parse-annotated Brown corpus only. This evaluation was performed for 2,713 active-verb lemmas using the more fine-grained Mapping-II. Tables 22 and 23 show that the results generally exceed the baseline, in some cases by almost 10%, similar to those recorded for Penn-II (Tables 17 and 18). While the precision is slightly lower than that reported for the experiments in Tables 17 and 18, in particular for Experiments 2, 2a, 3,

Table 21
Penn-II evaluation of active frames against COMLEX using absolute thresholds (Experiment 2).

Threshold	Precision	Recall	*F*-score
All	77.1%	50.4%	61.0%
Threshold 100	82.3%	60.8%	69.9%
Threshold 200	81.7%	58.7%	68.4%

Table 22
Results of Penn-III active frames (Brown Corpus only) COMLEX comparison (relative threshold of 1%).

	Precision		Recall		*F*-Score	
Mapping II	Baseline	Induced	Baseline	Induced	Baseline	Induced
Experiment 1	73.2%	79.2%	60.1%	60.0%	66.0%	68.2%
Experiment 2	66.0%	70.5%	37.5%	50.5%	47.8%	58.9%
Experiment 2a	66.0%	71.3%	32.7%	44.5%	43.7%	54.8%
Experiment 3	66.0%	64.3%	15.2%	23.1%	24.8%	34.0%
Experiment 3a	66.0%	64.1%	13.5%	20.7%	22.4%	31.3%

and 3a, in which details of obliques are included, the recall in each of these experiments is slightly higher than that recorded for Penn-II. We conjecture that the main reason for this is that 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. Consequently we record an increase in recall and a drop in precision. This trend is repeated in the second set of experiments. In this instance, we combined the lexicon extracted from the WSJ with that extracted from the parse-annotated Brown corpus, and evaluated the resulting resource for 3,529 active-verb lemmas. The results are shown in Tables 24 and 25. The results compare very positively against the baseline. The precision scores are lower (by between 1.5% and 9.7%) than those reported for Penn-II (Tables 17 and 18). There has however been a significant increase in recall (up to 8.7%) and an overall increase in F-score (by up to 4.4%).

6.5 Error Analysis and Discussion

The work presented in this section highlights a number of issues associated with the evaluation of automatically induced subcategorization frames against an existing external gold standard, in this case COMLEX. While this evaluation approach is arguably less labor-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 nontrivial task to convert both the gold standard and the induced resource to a common

Table 23
Results of Penn-III active frames (Brown corpus only) COMLEX comparison (relative threshold of 5%).

	Precision		Recall		*F*-score	
Mapping II	Baseline	Induced	Baseline	Induced	Baseline	Induced
Experiment 1	73.2%	82.7%	60.1%	56.4%	66.0%	67.0%
Experiment 2	66.0%	74.6%	37.5%	46.1%	47.8%	57.0%
Experiment 2a	66.0%	76.0%	32.7%	40.0%	43.7%	52.4%
Experiment 3	66.0%	69.2%	15.2%	18.7%	24.8%	29.5%
Experiment 3a	66.0%	69.0%	13.5%	16.6%	22.4%	26.7%

Table 24
Results of Penn-III active frames (Brown and WSJ) COMLEX comparison (relative threshold of 1%).

	Precision		Recall		*F*-score	
Mapping II	Baseline	Induced	Baseline	Induced	Baseline	Induced
Experiment 1	71.2%	77.4%	62.9%	66.2%	66.8%	71.4%
Experiment 2	64.5%	70.4%	40.0%	58.0%	49.3%	63.6%
Experiment 2a	64.5%	71.5%	35.1%	51.9%	45.5%	60.2%
Experiment 3	64.5%	66.2%	17.0%	27.4%	26.8%	38.8%
Experiment 3a	64.5%	66.0%	15.1%	24.8%	24.5%	36.0%

format in order to facilitate evaluation. In addition, as our results show, the choice of common format and mapping to it can affect the results. In COMLEX-LFG Mapping I (Section 6.2), we found that mapping from the induced lexicon to COMLEX resulted in higher recall scores than those achieved when we (effectively) reversed the mapping (COMLEX-LFG Mapping II [Section 6.3]). The first mapping is essentially a conflation of our more fine-grained LFG grammatical functions with the more generic COMLEX functions, while the second mapping tries to maintain as many distinctions as possible.

Another drawback to using an existing external gold standard such as COMLEX to evaluate an automatically induced subcategorization lexicon is that the resources are not necessarily constructed from the same source data. As noted above, it is well documented (Roland and Jurafsky 1998) that subcategorization frames (and their frequencies) vary across domains. We 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 subcategorization frames than our induced lexicon. It is also possible that because of human error, COMLEX contains subcategorization frames the validity of which are in doubt, for example, the overgeneration of subcategorized-for directional prepositional phrases. This is because the aim of the COMLEX project was to construct as complete a set of subcategorization frames as possible, even for infrequent verbs. Lexicographers were allowed to extrapolate from the citations found, a procedure

Table 25
Results of Penn-III active frames (Brown and WSJ) COMLEX comparison (relative threshold of 5%).

	Precision		Recall		*F*-score	
Mapping II	Baseline	Induced	Baseline	Induced	Baseline	Induced
Experiment 1	71.2%	82.0%	62.9%	61.0%	66.8%	69.9%
Experiment 2	64.5%	74.3%	40.0%	53.5%	49.3%	62.2%
Experiment 2a	64.5%	76.4%	35.1%	45.1%	45.5%	56.7%
Experiment 3	64.5%	71.1%	17.0%	21.5%	26.8%	33.0%
Experiment 3a	64.5%	70.8%	15.1%	19.2%	24.5%	30.2%

which is bound to be less certain than the assignment of frames based entirely on existing examples. As a generalization, Briscoe (2001) notes that lexicons such as COMLEX tend to demonstrate high precision but low recall. Briscoe and Carroll (1997) report on manually analyzing an open-class vocabulary of 35,000 head words for predicate subcategorization information and comparing the results against the subcategorization 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 our induced lexicon, we carried out some more-detailed error analysis, the results of which are summarized in Table 26. We 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 as "correct" or "incorrect." Of the 80 fps, 33 were manually judged to be legitimate subcategorization frames. For example, as Table 26 shows, there are a number of correct transitive verbs ([subj,obj]) in our automatically induced lexicon which are not included in COMLEX. This examination was also useful in highlighting to us 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 ditransitive frames, three frames containing a COMP, and seven frames containing an oblique were found to be invalid.

7. Lexical Accession Rates

In addition to evaluating the quality of our extracted semantic forms, we also examined the rate at which they are induced. This can be expressed as a measure of the coverage of the induced lexicon on new data. Following Hockenmaier, Bierner, and Baldridge (2002), Xia (1999), and Miyao, Ninomiya, and Tsujii (2004), we extract a reference lexicon from Sections 02–21 of the WSJ. We then compare this to a test lexicon from Section 23. Table 27 shows the results of the evaluation of the coverage of an induced lexicon for verbs only. There is a corresponding semantic form in the reference lexicon for 89.89% of the verbs in Section 23. 10.11% of the entries in the test lexicon did not appear in the reference lexicon. Within this group, we 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 we make the distinction

Table 26
Error analysis.

Frame type	COMLEX: False negatives		Induced: False positives	
	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 27
Coverage of induced lexicon (WSJ 02–21) on unseen data (WSJ 23) (verbs only).

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

between known frames and unknown frames. There are, therefore, four different cases in which an entry may not appear in the reference lexicon. Table 27 shows that the most common case is that of known verbs occurring with a different, although known, subcategorization frame (7.85%).

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. We were interested in discovering whether the acquisition of lexical material from the same data displayed a similar propensity. Figure 8 graphs the rate of induction of semantic form and CFG rule types from Penn-III (the WSJ and parse-annotated Brown corpus combined). Because of the variation in the size of sections between the Brown and the WSJ, we plotted accession against word count. The first part of the graph (up to 1,004,414 words)

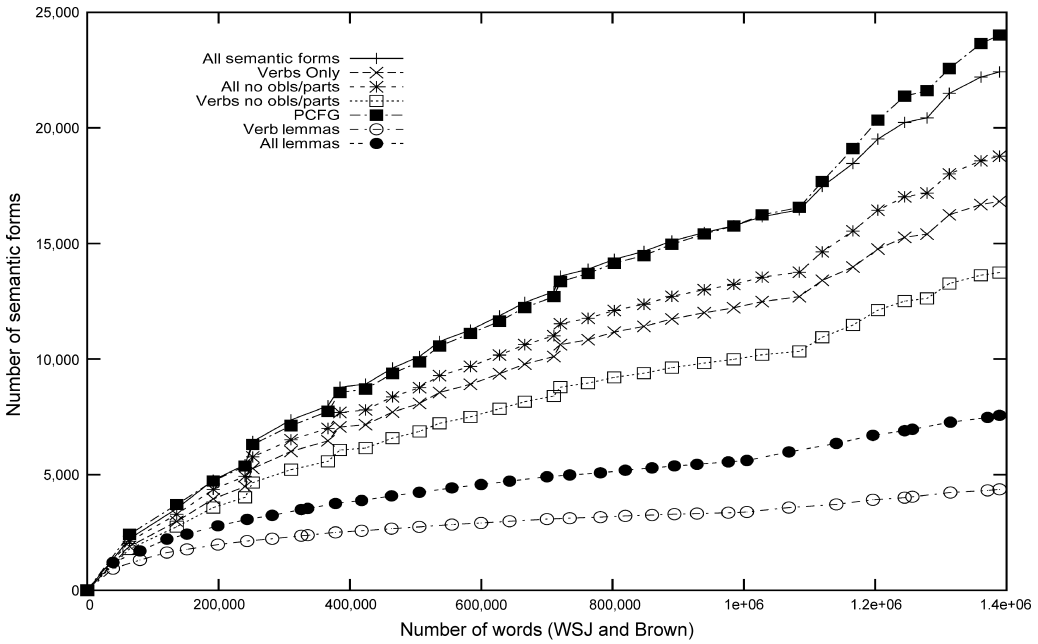


Figure 8
Comparison of accession rates for semantic form and CFG rule types for Penn-III (nonempty frames) (WSJ followed by Brown).

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 (nonempty) for all syntactic categories with and without specific preposition and particle information, the acquisition of semantic form types (nonempty) for all verbs with and without specific preposition and particle information, the number of lemmas associated with the extract semantic forms, and the acquisition of CFG rule types. The curve 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 8 shows the effect of domain diversity from the Brown section in terms of increased growth rates for 1e+06 words upward. Figure 9 depicts the same information, 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 8 and 9.

Figure 10 shows the result when we abstract away from semantic forms (verb frame combinations) to subcategorization 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 in Table 27, where we demonstrate that although new verb frame combinations occur, all of the frame types in Section 23 have been seen by the lexical extraction program in previous sections.

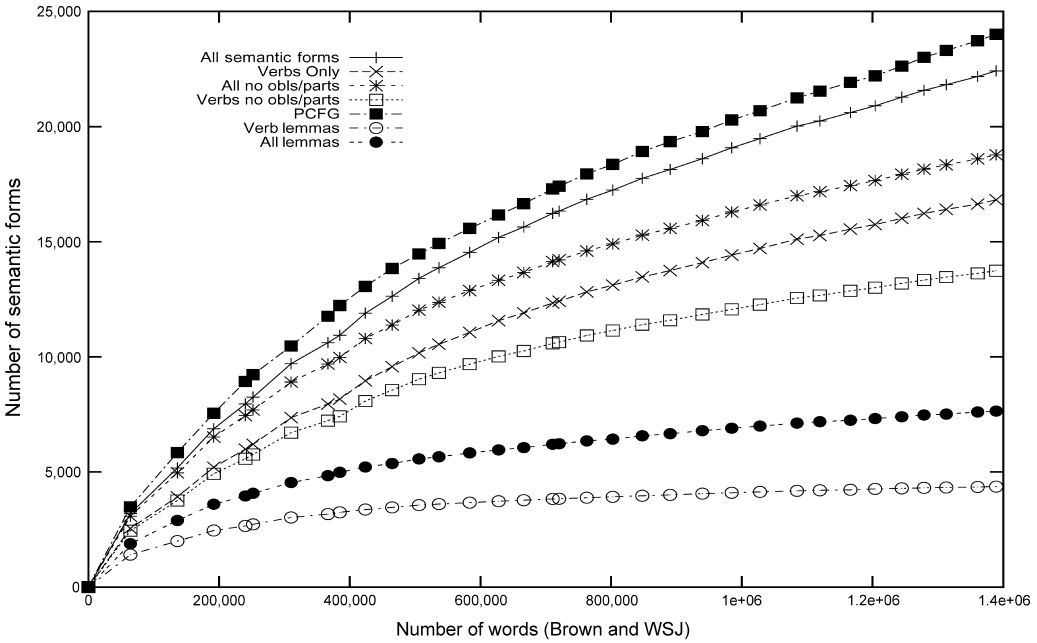


Figure 9
Comparison of accession rates for semantic form and CFG rule types for Penn-III (nonempty frames) (Brown followed by WSJ).

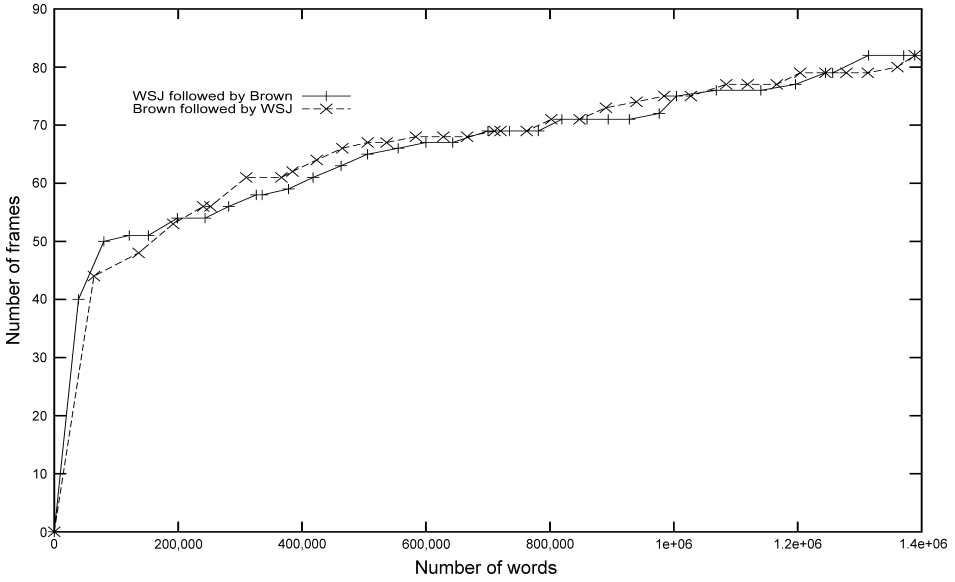


Figure 10
Accession rates for frame types (without prepositions and particles) for Penn-III.

Figure 11 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 emphasizes the advantage of our approach, which extracts frames containing such information without the limitation of predefinition.

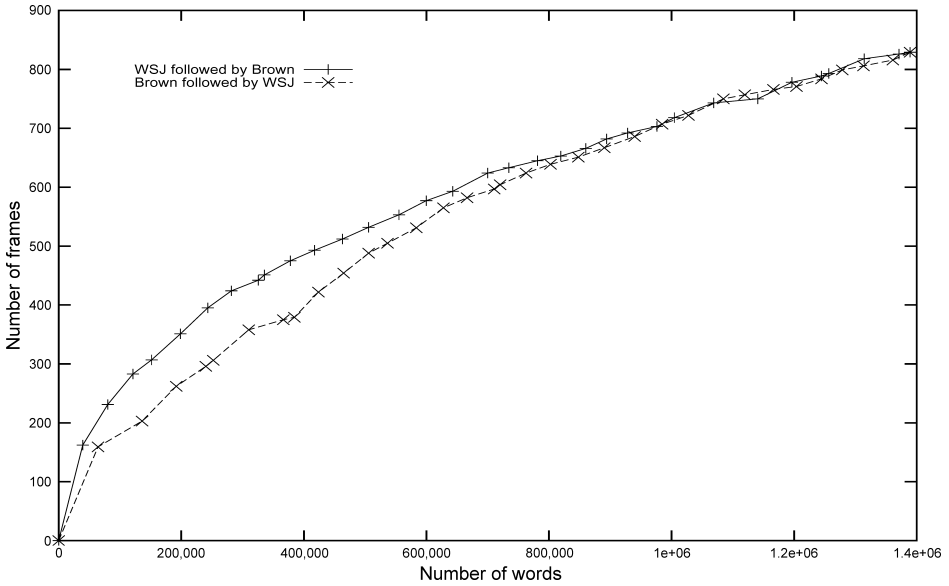


Figure 11
Accession rates for frame types for Penn-III.

8. Conclusions and Further Work

We have presented an algorithm for the extraction of semantic forms (or subcategorization frames) from the Penn-II and Penn-III Treebanks, automatically annotated with LFG f-structures. In contrast to many other approaches, ours does not predefine the subcategorization frames we extract. We have applied the algorithm to the WSJ sections of Penn-II (50,000 trees) (O'Donovan et al. 2004) and to the parse-annotated Brown corpus of Penn-III (almost 25,000 additional trees). We extract syntactic-function-based subcategorization frames (LFG semantic forms) and traditional CFG category-based frames, as well as mixed-function-category-based frames. Unlike many other approaches to subcategorization frame extraction, our system properly reflects the effects of long-distance dependencies. Also unlike many approaches, our method distinguishes between active and passive frames. Finally, our system associates conditional probabilities with the frames we extract. Making the distinction between the behavior of verbs in active and passive contexts is particularly important for the accurate assignment of probabilities to semantic forms. We carried out an extensive evaluation of the complete induced lexicon against the full COMLEX resource. To our knowledge, this is the most extensive qualitative evaluation of subcategorization extraction in English. The only evaluation of a similar scale is that carried out by Schulte im Walde (2002b) for German. The results reported here for Penn-II compare favorably against the baseline and, in fact, are an improvement on those reported in O'Donovan et al. (2004). The results for the larger, more domain-diverse Penn-III lexicon are very encouraging, in some cases almost 15% above the baseline. We believe our semantic forms are fine-grained, and by choosing to evaluate against COMLEX, we set our sights high: COMLEX is considerably more detailed than the OALD or LDOCE used for other earlier evaluations. Our error analysis also revealed some interesting issues associated with using an external standard such as COMLEX. In the future, we hope to evaluate the automatic annotations and extracted lexicon against Propbank (Kingsbury and Palmer 2002).

Apart from the related approach of Miyao, Ninomiya, and Tsujii (2004), which does not distinguish between argument and adjunct prepositional phrases, our treebank and automatic f-structure annotation-based architecture for the automatic acquisition of detailed subcategorization frames is quite unlike any of the architectures presented in the literature. Subcategorization frames are reverse-engineered and almost a byproduct of the automatic f-structure annotation algorithm. It is important to realize that the induction of lexical resources is part of a larger project on the acquisition of wide-coverage, robust, probabilistic, deep unification grammar resources from treebanks Burke, Cahill, et al. (2004b). We are already using the extracted semantic forms in parsing new text with robust, wide-coverage probabilistic LFG grammar approximations automatically acquired from the f-structure-annotated Penn-II treebank, specifically in the resolution of LDDs, as described in Cahill, Burke, et al. (2004). We hope to be able to apply our lexical acquisition methodology beyond existing parse-annotated corpora (Penn-II and Penn-III): New text is parsed by our probabilistic LFG approximations into f-structures from which we can then extract further semantic forms. The work reported here is part of the core components for bootstrapping this approach.

In the shorter term, we intend to make the extracted subcategorization lexicons from Penn-II and Penn-III available as a downloadable public-domain research resource.

We have also applied our more general unification grammar acquisition methodology to the TIGER Treebank (Brants et al. 2002) and Penn Chinese Treebank (Xue, Chiou, and Palmer 2002), extracting wide-coverage, probabilistic LFG grammar

approximations and lexical resources for German (Cahill et al. 2003) and Chinese (Burke, Lam, et al. 2004). The lexical resources, however, have not yet been evaluated. This, and much else, has to await further research.

Acknowledgments

The research reported here is partially supported by Enterprise Ireland Basic Research Grant SC/2001/186, an IRCSET PhD fellowship award, and an IBM PhD fellowship award. We are particularly grateful to our anonymous reviewers, whose insightful comments have helped to improve this article considerably.

References

Ades, Anthony and Mark Steedman. 1982. On the order of words. *Linguistics and Philosophy*, 4(4):517–558.

Boguraev, Branimir, Edward Briscoe, John Carroll, David Carter, and Claire Grover. 1987. The derivation of a grammatically indexed lexicon from the *Longman Dictionary of Contemporary English*. In *Proceedings of the 25th Annual Meeting of the Association of Computational Linguistics*, pages 193–200, Stanford, CA.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, Bulgaria.

Brent, Michael. 1993. From grammar to lexicon: Unsupervised learning of lexical syntax. *Computational Linguistics*, 19(2):203–222.

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Blackwell, Oxford.

Briscoe, Edward. 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*, Lancaster, UK.

Briscoe, Edward and John Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Proceedings of the Fifth ACL Conference on Applied Natural Language Processing*, pages 356–363, Washington, DC.

Burke, Michael, Aoife Cahill, Ruth O'Donovan, Josef van Genabith, and Andy Way. 2004a. 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, Michael, Aoife Cahill, Ruth O'Donovan, Josef van Genabith, and Andy Way. 2004b. Treebank-based acquisition of wide-coverage, probabilistic LFG resources: Project overview, results and evaluation. 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)*, Hainan Island, China.

Burke, Michael, Olivia Lam, Rowena Chan, Aoife Cahill, Ruth O'Donovan, Adams Bodomo, Josef van Genabith, and Andy Way. 2004. Treebank-based acquisition of a Chinese lexical-functional grammar. In *Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation*, pages 161–172, Tokyo.

Cahill, Aoife, Michael Burke, Ruth O'Donovan, Josef van Genabith, and Andy Way. 2004. Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based LFG approximations. In *Proceedings of the 42nd Annual Meeting of the Association of Computational Linguistics*, pages 320–327, Barcelona.

Cahill, Aoife, Martin Forst, Mairead McCarthy, Ruth O'Donovan, Christian Rohrer, Josef van Genabith, and Andy Way. 2003. Treebank-based multilingual unification-grammar development. In *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development at the 15th ESS-LLI*, pages 17–24, Vienna.

Cahill, Aoife, Mairead McCarthy, Michael Burke, Ruth O'Donovan, Josef van Genabith, and Andy Way. 2004. Evaluating automatic F-structure annotation for the Penn-II Treebank. *Journal of Research on Language and Computation*, 2(4):523–547.

Cahill, Aoife, Mairead McCarthy, Josef van Genabith, and Andy Way. 2002. Parsing text with a PCFG derived from Penn-II with an automatic F-structure annotation procedure. In *Proceedings of the Seventh International Conference on LFG*, edited by Miriam Butt and Tracy Holloway King. CSLI Publications, Stanford, CA, pages 76–95.

Carroll, Glenn and Mats Rooth. 1998. Valence induction with a head-lexicalised PCFG. In *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing*, pages 36–45, Granada, Spain.

Charniak, Eugene. 1996. Tree-bank grammars. In *AAAI-96: Proceedings of the Thirteenth National Conference on Artificial Intelligence*. MIT Press, Cambridge, MA, pages 1031–1036.

Chen, John and K. Vijay-Shanker. 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.

Collins, Michael. 1997. Three generative lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, Madrid.

Crouch, Richard, Ron Kaplan, Tracy King, and Stefan Riezler. 2002. A comparison of evaluation metrics for a broad coverage parser. In *Proceedings of Workshop "Beyond PARSEVAL" at Third International Conference on Language Resources and Evaluation*, Las Palmas, Spain.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*. Volume 34 of *Syntax and Semantics*. Academic Press, New York.

Dowty, David. 1982. Grammatical relations and Montague grammar. In Pauline Jacobson and Geoffrey Pullum, editors, *The Nature of Syntactic Representation*. Reidel, Dordrecht, The Netherlands, pages 79–130.

Dudenredaktion, editor. 2001. *DUDEN—Das Stilworterbuch*. [DUDEN—The Style Dictionary]. Number 2 in *Duden in zwölf Banden* [Duden in Twelve Volumes]. Dudenverlag, Mannheim, Germany.

Eckle, Judith. 1999. *Linguistic Knowledge for Automatic Lexicon Acquisition from German Text Corpora*. Ph.D. thesis, University of Stuttgart, Germany.

Frank, Anette. 2000. Automatic F-structure annotation of treebank trees. In *Proceedings of the Fifth International Conference on LFG*, Berkeley, CA, edited by Miriam Butt and Tracy Holloway King. CSLI, pages 139–160.

Grishman, Ralph, Catherine MacLeod, and Adam Meyers. 1994. COMLEX syntax: Building a computational lexicon. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 268–272, Kyoto.

Hajic, Jan. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In *Issues in Valency and Meaning*, edited by Eva Hajicova. Karolinum, Prague, Czech Republic, pages 106–132.

Hindle, Donald and Mats Rooth. 1993. Ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

Hockenmaier, Julia, Gann Bierner, and Jason Baldridge. 2004. Extending the coverage of a CCG system. *Journal of Language and Computation*, 2(2):165–208.

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

Joshi, Aravind. 1988. Tree adjoining grammars. In David Dowty, Lauri Karttunen, and Arnold Zwicky, editors, *Natural Language Parsing*. Cambridge University Press, Cambridge, pages 206–250.

Kaplan, Ronald and Joan Bresnan. 1982. Lexical functional grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA, pages 173–281.

King, Tracy Holloway, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald Kaplan. 2003. The PARC 700 Dependency Bank. In *Proceedings of the Fourth International Workshop on Linguistically Interpreted Corpora*, Budapest.

Kingsbury, Paul and Martha Palmer. 2002. From Treebank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, Spain.

Kinyon, Alexandra and Carlos Prolo. 2002. Identifying verb arguments and their syntactic function in the Penn Treebank. In *Proceedings of the Third LREC Conference*, pages 1982–1987, Las Palmas, Spain.

Korhonen, Anna. 2002. Subcategorization acquisition. As Technical Report UCAM-CL-TR-530, Computer Laboratory, University of Cambridge, UK.

Krotov, Alexander, Mark Hepple, Robert Gaizauskas, and Yorick Wilks. 1998. Compacting the Penn Treebank grammar. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 669–703, Montreal.

Levin, Beth. 1993. *English Verb Classes and Alternations*. University of Chicago Press, Chicago.

MacLeod, Catherine, Ralph Grishman, and Adam Meyers. 1994. The Comlex Syntax Project: The first year. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 669–703, Princeton.

Magerman, David. 1994. *Natural Language Parsing as Statistical Pattern Recognition*. Ph.D. thesis, Stanford University, Stanford, CA.

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

Manning, Christopher. 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, Mitchell, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the ARPA Human Language Technology Workshop*, Princeton.

Marinov, Svetoslav and Cecilia Hemming. 2004. Automatic Extraction of Subcategorization Frames from the Bulgarian Tree Bank. Unpublished manuscript, Graduate School of Language Technology, Göteborg, Sweden.

Meyers, Adam, Catherine MacLeod, and Ralph Grishman. 1996. Standardization of the complement/adjunct distinction. In *Proceedings of the Seventh EURALEX International Conference*, Göteborg, Sweden.

Miyao, Yusuke, Takashi Ninomiya, and Jun'ichi Tsujii. 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)*, pages 390–398, Hainan Island, China.

Nakanishi, Hiroko, Yusuke Miyao, and Jun'ichi Tsujii. 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)*, Hainan Island, China.

O'Donovan, Ruth, Michael Burke, Aoife Cahill, Josef van Genabith, and Andy Way. 2004. Large-scale induction and evaluation of lexical resources from the Penn-II Treebank. In *Proceedings of the 42nd Annual Meeting of the Association of Computational Linguistics*, pages 368–375, Barcelona.

Pollard, Carl and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Proctor, Paul, editor. 1978. *Longman Dictionary of Contemporary English*. Longman, London.

Roland, Douglas and Daniel Jurafsky. 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.

Sadler, Louisa, Josef van Genabith, and Andy Way. 2000. Automatic F-structure annotation from the AP Treebank. In *Proceedings of the Fifth International Conference on LFG*, Berkeley, CA, edited by Miriam Butt and Tracy Holloway King. CSLI, pages 226–243.

Sarkar, Anoop and Daniel Zeman. 2000. Automatic extraction of subcategorization frames for Czech. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 691–697, Saarbrücken, Germany.

Schulte im Walde, Sabine. 2002a. A subcategorisation lexicon for German verbs induced from a lexicalised PCFG. In *Proceedings of the Third LREC Conference*, pages 1351–1357, Las Palmas, Spain.

Schulte im Walde, Sabine. 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.

Simov, Kiril, Gergana Popova, and Petya Osenova. 2002. HPSG-based syntactic treebank of Bulgarian (BulTreeBank). In Andrew Wilson, Paul Rayson, and Tony McEnery, editors, *A Rainbow of Corpora: Corpus Linguistics and the Languages of the World*. Lincon-Europa, Munich, pages 135–142.

Ushioda, Akira, David Evans, Ted Gibson, and Alex Waibel. 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, Josef, Louisa Sadler, and Andy Way. 1999. Data-driven compilation of LFG semantic forms. In *EACL-99 Workshop on Linguistically Interpreted Corpora (LINC-99)*, pages 69–76, Bergen, Norway.

van Genabith, Josef, Andy Way, and Louisa Sadler. 1999. Semi-automatic generation of F-structures from Treebanks. In *Proceedings of the Fourth International Conference on Lexical-Functional Grammar,* Manchester, UK. Available at http://cslipublications.stanford.edu/.

Wauschkuhn, Oliver. 1999. *Automatische Extraktion von Verbvalenzen aus deutschen Textkorpora* [Automatic Extraction of Verb Valence from German Text Corpora]. PhD thesis, University of Stuttgart, Germany.

Xia, Fei. 1999. Extracting tree adjoining grammars from bracketed corpora. In *Fifth Natural Language Processing Pacific Rim Symposium (NLPRS-99)*, Beijing, China.

Xue, Nianwen, Fu-Dong Chiou, and Martha Palmer. 2002. Building a large-scale annotated Chinese corpus. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.