Treebank-Based Acquisition of Wide-Coverage, Probabilistic LFG Resources: Project Overview, Results and Evaluation

Michael Burke, Aoife Cahill, Ruth O'Donovan, Josef van Genabith, Andy Way
National Centre for Language Technology and School of Computing,
Dublin City University, Dublin, Ireland
{mburke,acahill,rodonovan,josef,away}@computing.dcu.ie

Abstract

This paper presents an overview of a project to acquire wide-coverage, probabilistic Lexical-Functional Grammar (LFG) resources from treebanks. Our approach is based on an automatic annotation algorithm that annotates "raw" treebank trees with LFG f-structure information approximating to basic predicate-argument/dependency structure. From the f-structure-annotated treebank we extract probabilistic unification grammar resources. We present the annotation algorithm, the extraction of lexical information and the acquisition of wide-coverage and robust PCFG-based LFG approximations including long-distance dependency resolution. We show how the methodology can be applied to multilingual, treebank-based unification grammar acquisition. Finally we show how simple (quasi-)logical forms can be derived automatically from the f-structures generated for the treebank trees.

1 Introduction

Manually scaling modern unification/constraint-based grammars such as LFG and HPSG to naturally occurring free text is very time-consuming, expensive and requires considerable linguistic and computational expertise. Few hand-crafted, deep grammars have in fact achieved the coverage and robustness required to parse a corpus of, e.g., the size and complexity of the Penn treebank: (Riezler *et al*, 2002) show how a carefully hand-crafted LFG is successfully scaled to parse the Penn-II treebank (Marcus et al, 1994) with discriminative (log-linear) parameter estimation techniques, providing deep linguistic analyses including long-distance dependencies (LDDs) and basic predicate-argument/dependency structure in the form of LFG f-structures.

The last 20 years have seen continuously increasing efforts in the construction of parse-annotated corpora. Substantial treebanks[1] are now available for many languages (including English, Japanese, Chinese, German, French, Czech, Turkish), others are currently under construction (Bulgarian) or near completion (Spanish). First generation treebanks represented mainly surface syntactic information in the form of CFG parse trees, while many second generation treebanks also include "deep" information such as LDDs, abstract syntactic function or dependency information. This information is often provided expressly to support the computation of meaning representations.

Treebanks have been enormously influential in the development of robust, state-of-the-art parsing technology: grammars (or grammatical information) automatically extracted from treebank resources provide the backbone of many state-of-the-art probabilistic parsing approaches (Charniak, 1996; Collins, 1999; Clark *et al.*, 2002; Klein and Manning, 2003). Such approaches are attractive as

[1] Or dependency banks.

they achieve robustness, coverage and performance while incurring very low grammar development cost. However, with very few notable exceptions (Collins model 3, Johnson (2002), CCG), treebank-based probabilistic parsers return fairly simple "surfacy" CFG trees, without deep syntactic (LDD) or semantic information. Because of the this, the grammars used by such systems are sometimes referred to as "half-grammars" (Johnson, 2002).

In this paper we provide an overview of a project that in many respects is a natural development and extension of the basic, automatic treebank PCFG acquisition paradigm (Charniak, 1996; Johnson, 1998; Klein and Manning, 2003). Rather than extracting simple CFGs, however, our aim is a treebank-based grammar acquisition methodology for deep, wide-coverage unification grammars such as LFG. We observed that many (second generation) treebanks provide a certain amount of deep syntactic information (e.g. Penn-II functional annotations and traces) supporting the computation of deep linguistic information. Exploiting this information we design and implement an automatic f-structure annotation algorithm that associates nodes in treebank trees with f-structure annotations in the form of attribute-value structure equations representing abstract predicate-argument structure/dependency relations. From the f-structure annotated treebank we automatically extract wide-coverage unification grammar resources (subcategorisation frames and PCFG-based LFG approximations) for parsing new text into f-structures.

Some aspects of our approach and early results have been published elsewhere (Cahill *et al.*, 2002; Cahill *et al.*, 2004a). This paper gives, for the first time, a comprehensive project overview, presents new results and project extensions not reported elsewhere. The paper is structured as follows: first, we give a brief introduction to LFG. Second, we detail the automatic f-structure annotation algorithm. Third, we present our subcategorisation frame extraction. Fourth, we present our PCFG-based LFG grammar approximations and parsing architectures. Fifth, while our original approach was developed for English and the Penn-II treebank, we show how it can be migrated successfully to a typologically different language (German) and treebank encoding (Brants et. al, 2002). Finally we show how simple (quasi-)logical forms can be derived from the f-structures generated for the treebank trees. In each case we provide results and extensive evaluation.

2 Lexical-Functional Grammar

Lexical-Functional Grammar (Kaplan and Bresnan, 1982; Bresnan, 2001; Dalrymple, 2001) minimally involves two levels of syntactic representation:[2] c-structure and f-structure. C(onstituent)-structure represents the grouping of words and phrases into larger constituents and is realised in terms of a CF-PSG grammar. F(unctional)-structure represents abstract syntactic functions such as SUBJ(ect), OBJ(ect), OBL(ique) argument, sentential and open COMP/XCOMP(lement), ADJ(unct), APP(osition) etc. and is implemented in terms of recursive feature structures (attribute-value matrices, AVMs). C-structure captures surface grammatical configurations, f-structure encodes abstract syntactic information approximating to predicate-argument-adjunct structure or simple logical form (van Genabith and Crouch, 1996). Alternatively, f-structures can be regarded as AVM encodings of dependency relations. C-structures and f-structures are related (non-transformationally) in terms of functional annotations (constraints, attribute-value equations) on c-structure rules (cf. Figure 1).

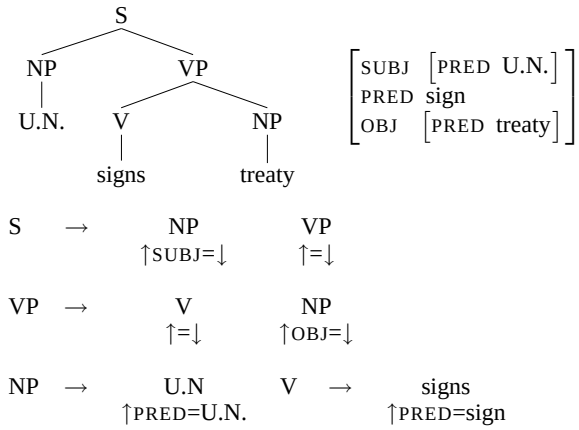

Figure 1: Simple LFG c- and f-structure

3 Automatic F-Structure Annotation

The Penn-II treebank employs CFG trees with additional "functional" node annotations (such as -LOC,

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

-TMP, -SBJ, -LGS, ...) as well as traces and coindexation (to indicate LDDs) as basic data structures. The f-structure annotation algorithm exploits configurational, categorial, "functional" and head information to annotate nodes with feature-structure equations. We adapt Magerman's (1994) scheme to automatically head-lexicalise the Penn-II trees. This partitions local subtrees of depth one (corresponding to CFG rules) into left and right contexts (relative to head). The annotation algorithm is modular with four components (Figure 2): left-right (L-R) annotation principles (e.g. leftmost NP to right of V head of VP type rule is likely to be an object etc.); coordination annotation principles (separating these out simplifies other components of the algorithm); traces (translates traces and coindexation in trees into corresponding reentrancies in f-structure (1 in Figure 3)); catch all and clean-up. Lexical information is provided via macros for POS tag classes.

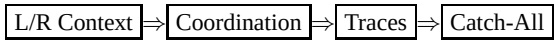

Figure 2: Annotation Algorithm

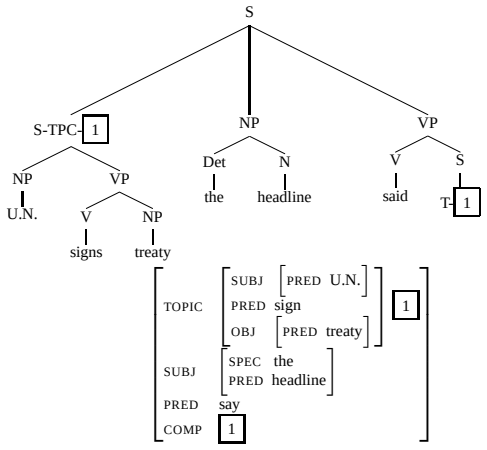

Figure 3: Penn-II style tree with LDD trace and corresponding reentrancy in f-structure

The f-structure annotations are passed to a constraint solver to produce f-structures. Annotation is evaluated in terms of coverage and quality, summarised in Table 1. Coverage is near complete with 99.83% of the 48K Penn-II sentences receiving a single, connected f-structure. Annotation quality is measured in terms of precision and recall (P&R) against manually constructed, gold-standard f-structures for 105 randomly selected trees from section 23 of the WSJ part of Penn-II. The algorithm currently achieves an F-score of 96.3% for complete f-structures and 93.6% for preds-only f-structures.[3]

# frags	# sent	percent
0	79	0.163
1	48343	99.833
2	2	0.004

	all	preds
P	96.07	93.38
R	96.44	93.97

Table 1: F-structure annotation results for Penn-II

A more detailed analysis of the f-structure quality is shown in Table 2. Precision and recall results are provided for selected features, e.g. the annotation algorithm achieves an f-score of 96% for the subject feature.

Feature	Precision	Recall	F-S
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

Table 2: Annotation results for selected features

4 Subcategorisation Frame Extraction

LFG distinguishes between governable (arguments) and nongovernable (adjuncts) grammatical functions (GFs). Subcategorisation requirements are stated in terms of GFs listed in what are referred to as "semantic forms" (subcategorisation frames). For example, the semantic form associated with transitive `see` is see([subj,obj]). If the automatic f-structure annotation algorithm outlined in Section 3 generates high quality f-structures, reliable semantic forms can be extracted (reverse-engineered) quite simply following (van Genabith et al., 1999): "For each f-structure generated, for each level of embedding we determine the local PRED value and collect the subcategorisable grammatical functions present

[3]Preds-only measures only paths ending in PRED:VALUE.

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

Table 3: Semantic forms for the verb accept.

at that level of embedding." For the f-structure in Figure 3 we obtain the non-empty semantic forms sign([subj,obj]) and say([subj,comp]).

We substantially extend and scale the approach of (van Genabith et al., 1999), which was small-scale and 'proof of concept' (100 trees). We extract frames from the full WSJ section of the Penn-II Treebank with 48K trees. Unlike van Genabith (1999) and many other approaches, our extraction process fully reflects LDDs, indicated in terms of traces in the Penn-II Treebank and corresponding re-entrancies at f-structure (cf. Figure 3). We do not predefine the frames to be extracted by our system. We discriminate between active and passive frames. We compute GF, GF:CFG category pair- as well as CFG category-based subcategorisation frames. Finally, we associate conditional probabilities with frames. Given a lemma l and an argument list s, the probability of s given l is estimated as:

$$\mathcal{P}(s|l) := \frac{count(l,s)}{\sum_{i=1}^{n} count(l,s_i)}$$

We use relative thresholding to filter possible error judgements by our system. Table 3 shows the attested semantic forms for the verb accept with their associated conditional probabilities. Note that were the distinction between active and passive not taken into account, the transitive occurrence of accept would have been assigned an unmerited probability. [4]

[4]Given this distinction and the computed conditional probabilities of frames, it is easy to condition frames on both lemma (π) and voice (v: active/passive):

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

	Without Prep/Part	With Prep/Part
Lemmas	3586	3586
Sem. Forms	10969	14348
Frame Types	38	577
Active Frame Types	38	548
Passive Frame Types	21	177

Table 4: Verb Results

Table 13 indicates the scale of our induced lexical resources. We extract non-empty semantic forms[5] for 3586 verb lemmas, resulting in 10969 unique verbal semantic form types (lemma followed by non-empty argument list). Including prepositions associated with the OBLs and particles, this number goes up to 14348, an average of 4.0 per lemma. The number of unique frame types (without lemma) is 38 without specific prepositions and particles, 577 with.

We carried out a comprehensive evaluation of the automatically acquired verbal semantic forms against the COMLEX Resource (MacLeod et al., 1994) for the 2992 (actively used) verb lemmas that both resources have in common. The manually-constructed COMLEX entries provide a gold standard against which we evaluate the automatically induced frames. No other approach for the extraction of English subcategorisation details conducts an evaluation of this scale. The largest is that of (Carroll and Rooth, 1998) who evaluated 200 selected verbs. Only the evaluation carried out by (Schulte im Walde, 2002) for German is in anyway comparable in scale to ours.

COMLEX lexical entries express subcategorisation information as a value of a :SUBC feature in terms of surface syntactic constituents. However, what makes the COMLEX resource particularly suitable for our evaluation is that each of the complement types which make up the value of the :SUBC feature is associated with a formal frame definition encoding the functional arguments required by the pred in question. Here we report on the evaluation of GF-based frames only. In order to carry out the evaluation we mapped COMLEX GFs to LFG GFs.

We carried out three experiments. **Experiment 1** excludes prepositional phrases: e.g. the frame [subj,obl:for] becomes [subj]. **Experiment**

[5]Non-empty semantic forms contain at least one subcategorised grammatical function.

	Threshold 1%			Threshold 5%		
	P	R	F-Score	P	R	F-Score
Exp. 1	79.0%	59.6%	68.0%	83.5%	54.7%	66.1%
Exp. 2	77.1%	50.4%	61.0%	81.4%	44.8%	57.8%
Exp. 2a	76.4%	44.5%	56.3%	80.9%	39.0%	52.6%
Exp. 3	73.7%	22.1%	34.0%	78.0%	18.3%	29.6%
Exp. 3a	73.3%	19.9%	31.3%	77.6%	16.2%	26.8%

Table 5: COMLEX Comparison (Threshold 1% and 5%)

	Precision	Recall	F-Score
Exp. 3	81.7%	40.8%	54.4%
Exp. 3a	83.1%	35.4%	49.7%

Table 6: COMLEX Comparison using p-dir (Threshold of 1%)

2 includes prepositional phrases suppressing prepositions: [subj,obl:for] becomes [subj,obl]. **Experiment 3** evaluates the frames in full detail, [subj,obl:for]. **Experiments 2a** and **3a** are the same as **Experiment 2** and **3**, except that they include the particle with each PART function. We use conditional probability thresholds to filter the selection of semantic forms. Table 5 shows experiments with threshold 1% and 5%. The effect of the threshold increase is obvious in that Precision goes up for each of the experiments while Recall goes down. Our results compare well with those of (Schulte im Walde, 2002) for German.

The low recall figure in the case of **Experiments 3** and **3a** can be accounted for by the fact that COMLEX errs on the side of overgeneration when it comes to preposition assignment (Grishman *et al.*, , 1994). This is particularly true of directional prepositions, a list of 31 of which has been prepared and is assigned in its entirety by default to any verb which can potentially appear with any directional preposition. In a subsequent experiment, we incorporate 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 6 shows the significant improvement in recall for this experiment.

Table 7 presents the results of our evaluation of the passive semantic forms we extract. It was carried out for 1422 verbs which occur with passive frames and are shared by the induced lexicon and COMLEX. We applied Lexical Redundancy Rules (Kaplan and Bresnan, 1982) to automatically convert the active

Passive	Precision	Recall	F-Score
Exp. 2	80.2%	54.7%	65.1%
Exp. 2a	79.7%	46.2%	58.5%
Exp. 3	72.6%	33.4%	45.8%
Exp. 3a	72.3%	29.3%	41.7%

Table 7: Passive evaluation (Threshold of 1%)

COMLEX frames to their passive counterparts. The resulting precision was very high and there was the expected drop in recall when prepositional details were included.

We tested the coverage of our system in a similar way to (Hockenmaier et al., 2002). We extracted a verb-only reference lexicon from Sections 02-21 of the WSJ. We then compared this to a test lexicon constructed in the same way from Section 23. 89.89% of the entries in the test lexicon appeared in the reference lexicon.

5 PCFG-Based LFG Approximations

Based on the resources described above we developed two parsing architectures. Both generate PCFG-based approximations of LFG grammars.

In the **pipeline** architecture we first extract a standard PCFG from the "raw" treebank to parse unseen text. The resulting parse-trees are then annotated by the automatic f-structure annotation algorithm and resolved into f-structures. In the **integrated** architecture we first automatically annotate the treebank with f-structure equations. We then extract an annotated PCFG where each non-terminal symbol in the grammar has been augmented with LFG f-equations: NP[↑SUBJ=↓] → DT[↑SPEC=↓] NN[↑=↓] We treat a node followed by annotations as a monadic category for grammar extraction and parsing. Post-parsing, equations are collected from parse trees and resolved into f-structures. Both architectures parse raw text into "proto" f-structures with LDDs unresolved.

In LFG, LDDs are resolved at the f-structure level, obviating the need for empty productions and traces in trees (Dalrymple, 2001), using functional uncertainty (FU) equations. FUs are regular expressions specifying paths in f-structure between a source (where linguistic material is encountered) and a target (where linguistic material is interpreted

semantically). We *approximate* FUs by extracting paths between co-indexed material occurring in the automatically generated f-structures from sections 02-21 of the Penn-II treebank. We extract 26 unique TOPIC, 60 TOPIC-REL and 13 FOCUS path types, each with an associated probability. Given a path p and an LDD type t (either TOPIC, TOPIC-REL or FOCUS), the probability of p given t is estimated as:

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

For, say, a topicalised constituent to be resolved as the argument of a predicate as specified by a FU equation, the local predicate must (i) subcategorise for the argument and (ii) the argument must not be already filled locally. Subcategorisation requirements are provided by semantic forms as acquired in Section 4. The LDD resolution algorithm traverses an f-structure along LDD paths and ranks possible resolutions by multiplying path and semantic form probabilities involved. It supports multiple, interacting TOPIC, TOPIC-REL and FOCUS LDDs.

Given a set of semantic forms s with probabilities $\mathcal{P}(s|l)$ (where l is a lemma), a set of paths p with $\mathcal{P}(p|t)$ (where t is either TOPIC, TOPIC-REL or FOCUS) and an f-structure f, the core of the algorithm to resolve LDDs recursively traverses f to:

> find TOPIC|TOPIC-REL|FOCUS:g pair; retrieve TOPIC|TOPIC-REL|FOCUS paths; for each path p with $\text{GF}_1 : \ldots : \text{GF}_n : \text{GF}$, traverse f along $\text{GF}_1 : \ldots : \text{GF}_n$ to sub-f-structure h; retrieve local PRED:l;
>
> > add GF:g to h iff
> >
> > * GF is not present at h
> > * h together with GF is locally complete and coherent with respect to a semantic form s for l
> >
> > rank resolution by $\mathcal{P}(s|l) \times \mathcal{P}(p|t)$

We ran experiments with grammars in both the pipeline and the integrated parsing architectures. The first grammar is a basic PCFG, while A-PCFG includes the f-structure annotations. We train on sections 02-21 (grammar, lexical extraction and LDD paths) of the Penn-II Treebank and test on section 23. We evaluate the parse trees using evalb. Following (Riezler *et al*, 2002), we convert f-structures into dependency triple format. Using their software we evaluate the f-structure parser output against (i) *manually* constructed gold-standard f-structures for 105 randomly selected sentences from section 23 and (ii) in a CCG-style (Hockenmaier, 2003) evaluation experiment against the full 2,416 f-structures *automatically* generated by the f-structure annotation algorithm for the *original* Penn-II trees. The results are given in Table 8. The integrated model outperforms the pipeline model in all aspects of our evaluation. In all cases, there is a marked improvement (2.8-5.7%) in the f-structures after LDD resolution. We achieve between 78.2 and 79.1% preds-only and 83.9 to 84.9% all GF f-score, depending on gold-standard, with the integrated model. In order to measure how many of the LDD reentracies in the gold-standard f-structures are captured correctly by our parsers we developed evaluation software for f-structure LDD reentracies (similar to Johnson's (2002) evaluation to capture traces and their antecedents in trees). Table 9 show the results, with the integrated model achieving more than 79% correct LDD reentrancies.

	Pipeline PCFG	Integrated A-PCFG
	2416 Section 23 trees	
# Parses	2410	2398
Lab. F-Score	75.46	80.01
Unlab. F-Score	77.91	81.95
	105 F-Strs – pre LDD resolution	
All GFs	74.2	79.23
Preds only	64.75	73.19
	105 F-Strs – post LDD resolution	
All GFs	78.67	**84.93**
Preds only	69.23	**79.14**
	2416 F-Strs – pre LDD resolution	
All GFs	76.75	80.73
Preds only	67.5	75
	2416 F-Strs – post LDD resolution	
All GFs	79.51	**83.87**
Preds only	70.23	**78.24**

Table 8: Parser Evaluation

Our parsing architectures provide wide-coverage, robust, and - with the addition of LDD resolution - "deep" or "full", PCFG-based LFG *approximations*. We do not claim to provide fully adequate statistical models. It is well known (Abney, 1997) that PCFG-type approximations of full unification grammars can yield inconsistent probability models due to loss of probability mass: the parser successfully returns the highest ranked parse tree but the constraint solver cannot resolve the f-equations (generated in the pipeline or "hidden" in the integrated

	Pipeline PCFG	Integrated A-PCFG
TOPIC		
precision	12/13	13/13
recall	12/13	13/13
f-score	92.31	1
FOCUS		
precision	0/0	0/1
recall	0/1	0/1
f-score	0	0
TOPICREL		
precision	19/32	29/35
recall	19/43	29/43
f-score	50.67	74.36
OVERALL	60.78	**79.25**

Table 9: LDD Evaluation on 105 Gold Standard F-Structures

model) and the probability mass associated with that tree is lost. This case, however, is surprisingly rare for our grammars: only 0.0016% (79 out of 48424) of the original Penn-II trees (without FRAGs) fail to produce an f-structure due to inconsistent annotations (Table 1), and, e.g., for parsing section 23 with the integrated model 24 sentences do not receive a parse, 18 of which because there is no spanning tree, 6 because no f-structure can be generated for the highest ranked tree (0.25%).

6 Multilingual Grammar Acquisition

A number of strategies have been developed for multilingual, wide-coverage grammar development. The ParGram project (Butt *et al.*, 2002), e.g., develops wide-coverage (English, German, French, Japanese, ...) LFG grammars with harmonised feature inventories and geometries. Grammar migration can sometimes be used to seed grammars for typologically similar languages, such as e.g. Japanese and Korean (Kim *et al.*, 2003).

If treebank resources are available, treebank-based unification grammar acquisition can be a valuable alternative to traditional multilingual grammar development. We applied our methodology to German and the TIGER (Brants et. al, 2002) treebank. German is typologically different from English, in that it is much less configurational. The TIGER treebank is a corpus of approximately 40,000 syntactically annotated German newspaper sentences. The TIGER treebank data structures differ substantially from those in Penn-II: TIGER uses graphs with crossing edges (rather than trees with traces) to indicate non-local dependencies and provides considerably richer functional annotations.

We first convert TIGER graphs into Penn-II style trees with traces and coindexation encoding LDDs. There are two main components in the algorithm to automatically add f-structure information to TIGER trees, with a pre- and post-processing stage. The preprocessing is a simple walk through the tree in order to build a lookup table for the trace nodes. The first stage of the algorithm exploits the rich TIGER functional annotations by associating default f-structure equations with functional tags. Overgeneralisation is corrected in a second component (mainly with respect to adpositional case marking, complementisers and multiple coordinations). Finally, a post-processing stage explicitly links trace nodes and the reference node, encoding LDDs.

We evaluate the f-structures produced both qualitatively and quantitatively. Table 10 illustrates the coverage of the algorithm. Ideally we would like to generate just one f-structure per sentence. There are however, a number of sentences that receive more than one f-structure fragment. This is mainly due to sentences such as *Bonn, 7. September*, where in the source TIGER graphs there is no clear relation between the elements of the "sentence" and where we do not wish to enforce a relation for the sake of having fewer fragments. We believe that these "sentences" are in fact fragments and should be treated accordingly. There are also a small number of sentences which do not receive any f-structure. This is as a result of feature clashes in the annotated trees, most of which are caused by inconsistent annotations. We also evaluate the quality of the annotation against our manually constructed gold-standard of 100 sentences. Table 11 shows that currently our automatic annotation receives an f-score of 91.03%.

We extracted an annotated grammar (A-PCFG) from the TIGER corpus (excluding 2000 sentences set aside for testing). We parsed the 2000 raw sentences using Helmut Schmid's BitPar parser (p.c.). The results are presented in Table 12. We evaluate the quality of the trees produced by the parser with results of f-score 67.91 (Labelled) and 72.63 (Unlabelled). 95.9% of the 2000 sentences produce one complete f-structure (Fragmentation). We evaluate the quality of the f-structures produced in

# f-str. frags	# sent	percent
0	143	0.3573
1	38765	96.864
2	1032	2.5787
3	75	0.1874
5	1	0.0025
6	1	0.0025
7	3	0.0075

Table 10: Coverage & fragmentation results of German Annotation Algorithm

	Preds Only Evaluation
Precision	93.62
Recall	88.59
F-Score	91.03
Complete Match	25

Table 11: Evaluation of the f-structures produced by automatically annotating the TIGER trees

two ways. First we evaluate against our 100 manually constructed gold standard f-structures, achieving an f-score of 68.5. Second, in a manner similar to (Hockenmaier, 2003), we automatically annotate the 2000 trees to produce f-structures, and evaluate the output of the parser against these *automatically* produced f-structures. This experiment returns an f-score 62.62.

	A-PCFG
Lab. F-Score (2000 Trees)	67.91
Unlab. F-Score (2000 Trees)	72.63
Fragmenation (2000 f-structures)	95.9
F-Score(100 f-structures)	68.5
F-Score(2000 f-structures)	62.62

Table 12: Parsing Results

We automatically induce German subcategorisation frames using the methodology outlined in Section 4. As Table 13 shows, we extract 8632 non-empty semantic form types, 7081 of which are for verbs. (We extract frames for 4331 verb lemmas.) When the obliques are parameterised for prepositions we found an average of 1.63 semantic forms per verb. As for English, we also associate conditional probabilities with the frames extracted. Table 14 illustrates the most frequent subcategorisation frames for the verb aufhören with their associated conditional probabilities. We tested the coverage of our system in a similar way to that described for English in Section 4. We extract a reference verb

Sem. Form Types	8632
Verbs Only	7081

Table 13: Non-empty Semantic Forms extracted

Semantic Form	**Conditional Probability**
`aufhören([subj])`	0.444
`aufhören([subj,xcomp])`	0.389
`aufhören([subj,obl:mit])`	0.111
`aufhören([comp,subj])`	0.056

Table 14: Semantic Forms with associated conditional probability for the verb `aufhören`

lexicon from trees 1–8000 and 10001–40020 of the Tiger Treebank. We then compare this to a test lexicon from trees 8001–10000. For 86.75% of the entries in the test lexicon the corresponeding semantic form exists in the reference lexicon.

7 Quasi-Logical Forms

F-structures encode mainly abstract syntactic information with some semantic information in the form of basic predicate-argument structure. Quasi-logical forms, the semantic representation formalism of the Core Language Engine (Alshawi & Crouch, 1992), encode predominantly semantic with some syntactic information (syntactic information is represented as it constrains e.g. anaphora resolution and quantifier scope possibilities). Despite clear differences in approach and emphasis, unresolved QLFs and f-structures bear a striking similarity and, for simple cases at least, it is easy to see how to get from one to the other in terms of a translation function $(\cdot)^\circ$, cf. (van Genabith and Crouch, 1996):

$$\begin{bmatrix} \Gamma_1 \quad \gamma_1 \\ \dots \\ \text{PRED} \;\; \Pi\langle \uparrow \Gamma_1, ., \uparrow \Gamma_n \rangle \\ \dots \\ \Gamma_n \quad \gamma_n \end{bmatrix}^\circ \quad = \quad \texttt{?Scope:}\Pi(\gamma_1^\circ, ., \gamma_n^\circ)$$

The core of the $(\cdot)^\circ$ mapping taking us from f-structures to QLFs places the values of subcategorisable grammatical functions into their argument positions in the governing semantic form Π and recurses on those arguments. From this rather general perspective, the difference between f-structures and QLF is one of information packaging and presentation rather than anything else.

In (Cahill *et al.*, 2004b) we substantially extend the coverage of the more theoretically oriented work of (van Genabith and Crouch, 1996) to include passive constructions, wh-questions, relative clauses, fronted material and subjects of participal clauses, gerunds and infinitival clauses, modification (adjectival, adverbial, prepositional, appositional and sentential and non-sentential adjuncts as well as relative clauses) and coordinate/subordinate constructions:

```
An agreement was brokered by the U.N.
```

$$\begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`AGREEMENT'} \\ \text{NUM} & \text{SG} \\ \text{SPEC} & \text{A} \end{bmatrix} \\ \text{PRED} & \text{`BROKER}\langle\uparrow\text{SUBJ},\uparrow\text{OBJ}_{ag}\rangle\text{'} \\ \text{PASSIVE} & + \\ \text{OBL}_{ag} & \begin{bmatrix} \text{PRED} & \text{`BY}\langle\uparrow\text{OBJ}\rangle\text{'} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`U.N.'} \\ \text{NUM} & \text{SG} \\ \text{SPEC} & \text{THE} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

```
?Scope:broker(term(+r,<num=sg,spec=the>,
                   U.N.,?Q,?S),
              term(+g,<num=sg,spec=a>,
                   agreement,?P,?R))
```

Currently, the f-structure–QLF translation algorithm associates 95.76% (46371) of the 48424 trees (without FRAG and X constituents) in the Penn-II treebank with a QLF based on the f-structure automatically generated for the tree.

8 Conclusions

We have presented a method for treebank-based, wide-coverage, deep unification grammar acquisition. The resulting PCFG-based LFG approximations parse unseen Penn-II treebank English newspaper or unseen TIGER German newspaper text with wider coverage and results similar to those achieved by the best hand-crafted grammars, with, we believe, considerably less development effort. We have extracted wide-coverage lexical resources from the f-structure annotated treebanks and showed how simple (quasi-)logical forms can be derived from the f-structures generated. We believe that the approach (with suitable adaptations for HPSG, PATR and other constraint/unification based formalisms) can provide an attractive, wide-coverage, multilingual unification grammar acquisition paradigm, complementing and, in certain cases, replacing more traditional, manual grammar development.

Acknowledgments

The work reported in the paper was funded by Enterprise Ireland Basic Research Grant SC/2001/186 and by an IRCSET Ph.D. studentship.

References

Abney, S. P.; 1997; *Stochastic attribute-value grammars*; *Computational Linguistics*, 23(4)

Alshawi, H. and R. Crouch. (1992) Monotonic Semantic Interpretation, In *Proceedings 30th Annual Meeting of the Association for Computational Linguistics*, pages 32–38

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

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

Butt, M., H. Dyvik, T.H. King, H. Masuichi and C. Rohrer, 2002. 'The parallel grammar project.' *Proceedings of COLING 2002, Workshop on Grammar Engineering and Evaluation* pp. 1-7.

Cahill, A., M. McCarthy, J. van Genabith and A. Way; 2002. *Parsing with PCFGs and Automatic F-Structure Annotation.* In: M. Butt and T. Holloway-King (eds.) *Proceedings of the Seventh International Conference on LFG*, CSLI Publications, Stanford, CA., pp.76–95.

Cahill, A., M. McCarthy, M. Burke, R. O'Donovan, J. van Genabith and A. Way; 2004a. *Evaluating Automatic F-Structure Annotation for the Penn-II Treebank*. In: *Journal of Research on Language and Computation*, Kluwer Academic Publishers; in press.

Cahill, A., M. McCarthy, M. Burke, J. van Genabith and A. Way; 2004b. *Deriving Quasi-Logical Forms from F-Structures for the Penn Treebank.* In: H. Bunt and R. Muskens (eds.) *Computing Meaning, Vol-3*, Kluwer Academic Publishers; in press.

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

Charniak, E. 1996. Tree-bank Grammars. in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI Press/MIT Press, Menlo Park, pp.1031–1036

Clark, S., J. Hockenmaier and M. Steedman, 2002. Building Deep Dependency Structures with a Wide-Coverage CCG Parser Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02) Philadelphia, PA.

Collins, M. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia.

Dalrymple, M. 2001. *Lexical-Functional Grammar*. San Diego, Calif.; London : Academic Press

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

Hockenmaier, J., G. Bierner and J. Baldridge. 2002. Extending the Coverage of a CCG System. *Journal of Language and Computation*

Hockenmaier, J. 2003. Parsing with Generative models of Predicate-Argument Structure In *Proceedings of the 41st Annual Conference of the Association for Computational Linguistics (ACL-03)*, Sapporo, Japan

Johnson, M. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

Johnson, M. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.

Kaplan, R. and J. Bresnan 1982. Lexical-functional grammar: a formal system for grammatical representation. In Bresnan, J., editor 1982, *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge Mass. 173–281.

Kim, R., M. Dalrymple, R.M. Kaplan, T. Holloway King, H. Masuichi and T. Ohkuma. 2003. 'Multilingual Grammar Development via Grammar Porting.' In *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development*, Vienna, Austria.

Klein, D. and C.D. Manning, 2003. Accurate Unlexicalized Parsing Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'02) Sapporo, Japan

MacLeod, C., R. Grishman and A. Meyers. 1994. The Comlex Syntax Project: The First Year. In *Proceedings of the ARPA Workshop on Human Language Technology*, Princeton, NJ.

Magerman, D. 1994. Natural Language Parsing as Statistical Pattern Recognition PhD Thesis, Stanford University, CA.

Marcus, M., G. Kim, M. A. Marcinkiewicz, R. MacIntyre, M. Ferguson, K. Katz and B. Schasberger 1994. The Penn Treebank: Annotating Predicate Argument Structure. In: *Proceedings of the ARPA Human Language Technology Workshop*.

Riezler, S. T.H. King, R.M. Kaplan, R.Crouch, J.T. Maxwell, and M. Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02) Philadelphia, PA.

Schulte im Walde, S. 2002. 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*, Copenhagen, Denmark.

Van Genabith, J., A. Way and L. Sadler. 1999. Data-driven Compilation of LFG Semantic Forms. In *EACL-99 Workshop on Linguistically Interpreted Corpora*, Bergen, Norway, pp. 69–76.

Van Genabith, J. and D. Crouch 1996. Direct and Underspecified Interpretations of LFG f-Structures. In: *COLING 96*, Copenhagen, Denmark, Proceedings of the Conference. 262–267.