

Treebank-Based Multilingual Unification-Grammar Development

Aoife Cahill*, Martin Forst[†], Mairead McCarthy*, Ruth O'Donovan*,
Christian Rohrer[†], Josef van Genabith*, Andy Way*

*School of Computer Applications
Dublin City University
Dublin 9, Ireland

[†]Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
D-70174 Stuttgart, Germany

Abstract

Broad-coverage, deep unification grammar development is time-consuming and costly. This problem can be exacerbated in multilingual grammar development scenarios. Recently (Cahill *et al.*, 2002) presented a treebank-based methodology to semi-automatically create broad-coverage, deep, unification grammar resources for English. In this paper we present a project which adapts this model to a multilingual grammar development scenario to obtain robust, wide-coverage, probabilistic Lexical-Functional Grammars (LFGs) for English and German via automatic f-structure annotation algorithms based on the Penn-II and TIGER treebanks. We outline our method used to extract a probabilistic LFG from the TIGER treebank and report on the quality of the f-structures produced. We achieve an f-score of 66.23 on the evaluation of 100 random sentences against a manually constructed gold standard.

1 Introduction

Parsing is an important step in natural language processing as syntactic structure is a prime determinant for semantic interpretation in the form of predicate-argument structure, deep dependency structure or logical form. Rich unification (or rather: constraint) grammars such as LFG or HPSG model both (morpho-) syntactic and semantic information.

Manually scaling deep unification grammars to real text (such as the Penn-II treebank, Marcus *et al.*, 1994), however, is extremely time-consuming, costly and requires considerable linguistic and processing expertise, as it involves person-years of concerted grammar, lexicon and system (processing platform) development effort. What is more, few hand-crafted grammars achieve full coverage of the target corpus. Indeed, the only hand-crafted, deep unification grammar scaled to the full Penn-II treebank we are aware of is the English LFG grammar developed as part of the ParGram project at Xerox PARC (Riezler *et al.*, 2002; Butt *et al.*, 2002). This situation, we suspect, is even worse for languages other than English, as they have received considerably less (linguistic and computational linguistic) attention. Accordingly resource problems can be exacerbated in multilingual grammar development scenarios.¹

Recently (Cahill *et al.*, 2002) presented a treebank-based methodology to semi-automatically create wide-coverage, deep, unification LFG grammar resources for English. The method is based on an automatic annotation algorithm that annotates Penn-II treebank trees with LFG f-structure information. F-Structures are recursive attribute-value structures approximating to predicate-argument-modifier representations. Alternatively, f-structures can be viewed as encodings of deep dependency re-

¹Sometimes multilingual grammar development offers unique advantages unavailable to monolingual grammar development: these are cases where an existing grammar for language A can be migrated to a closely related language B for which such resources do not exist and thus significantly boost grammar development for language B. (Gamon *et al.*, 1997)

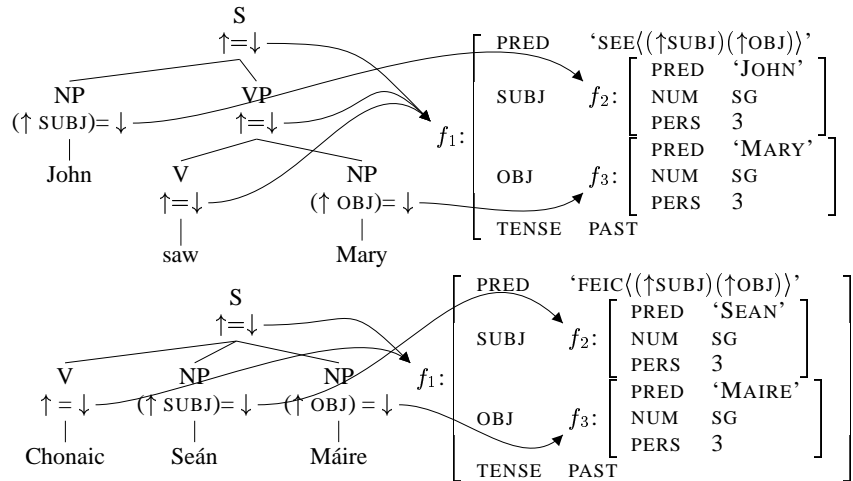


Figure 1: C- and f-structures for an English and corresponding Irish sentence

lations. (Cahill *et al.*, 2002) show how based on the f-structure annotated version of Penn-II, wide-coverage, robust, probabilistic LFG grammars can be derived to parse Penn-II data.

In this paper we first introduce LFG and, in particular, the level of f-structure representation and motivate why we think LFG provides a suitable representation format for multilingual grammar development. Next we present the basic ideas underlying the approach presented in (Cahill *et al.*, 2002) as applied to English. We outline how this approach can be adapted and migrated to a different language and treebank resource, namely German and the TIGER treebank (Brants *et al.*, 2002). German is substantially less configurational than English, and the TIGER treebank annotation consists of graphs with crossing edges rather than trees with traces (as in Penn-II). In addition, the TIGER treebank features considerably richer functional annotations than those provided in the Penn-II resource. We outline how LFG grammars for German can be derived from the f-annotated TIGER resource. We extract a probabilistic LFG and run some parsing experiments. We evaluate both the quality of the automatic annotation of the treebank, and the output of the parser which uses the probabilistic LFG. Finally, we present current work on automatic extraction of lexical resources from the f-structure annotated Penn-II treebank and recent work on long-distance dependencies. We then conclude and outline further work.

2 Lexical-Functional Grammar

Lexical-Functional Grammar (Kaplan and Bresnan, 1982; Bresnan, 2001) is an early member of the family of unification (or constraint) grammars (such as FUG, PATR-II, GPSG, CUP or HPSG). Minimally, LFG involves two levels of representation: c(onstituent)-structure and f(unctional)-structure. C-structure captures language specific phenomena such as word order and the grouping of constituents into larger phrases in the form of context-free trees. F-structure represents abstract syntactic functions such as subj(ect), obj(ect), pred(icate) etc. in the form of recursive attribute-value structures. The basic idea is that while languages may differ markedly with respect to surface realisation (c-structure), they may still exhibit very similar abstract syntactic functional representations (f-structure). Figure 1 illustrates this point. Irish is typologically a VSO-language, while English is an SVO-language. The same proposition expressed in Irish and English exhibits markedly different c-structure configurations but is associated with isomorphic (up to the values of PRED nodes) f-structure representations.

C-structure and f-structure representations are related in terms of "functional annotations" of the form $\uparrow \dots = \downarrow \dots$ to tree nodes, i.e. attribute-value structure equations (or more generally: disjunctive, implicational and negative constraints) describing f-structures.

F-structures approximate to predicate-argument-modifier representations, simple logical forms (van Genabith and Crouch, 1996) or deep dependency relations. LFG is particularly attractive for multilingual grammar development as the level of f-structure representation abstracts away from language-specific surface realisation (Butt *et al.*, 1999). At the same time LFG provides a precise, flexible, computationally tractable and non-transformational interface between c-structure and f-structure representation for both parsing and generation (Butt *et al.*, 2002). Unlike other unification grammar formalisms, LFG has enjoyed a substantial body of work on automatic f-structure annotation architectures summarised in (Cahill *et al.*, 2002; Frank *et al.*, 2003). These approaches automatically annotate (treebank or parse-generated) trees with f-structure equations to generate f-structures for those trees.

3 F-structures from Penn-II

(Cahill *et al.*, 2002) present an automatic f-structure annotation algorithm for the trees in the Penn-II treebank. They show how wide-coverage, robust probabilistic LFGs can be derived automatically. Given a tree, the task of f-structure annotation is to annotate tree nodes automatically with f-equations. As a simple example, consider the following CFG rule (i.e. a local tree of depth 1):

$$\text{NP} \rightarrow \text{DT ADJP NN SBAR}$$

Such a configuration would be associated with f-structure annotations as follows:

$$\begin{array}{cccccc} \text{NP} \rightarrow & \text{DT} & \text{ADJP} & \text{NN} & \text{SBAR} & \\ & \uparrow \text{SPEC} = \downarrow & \downarrow \in \uparrow \text{ADJ} & \uparrow = \downarrow & \downarrow \in \uparrow \text{RELMOD} & \end{array}$$

This indicates that the NN is the head of the NP, the DT is a specifier, the adjective phrase is part of the modifying adjunct set, and the SBAR is a member of the RELMOD set.

The annotation algorithm automatically transforms trees into head-lexicalised trees using a variant of (Magerman, 1994)’s head rules and then uses configurational, categorial, functional tag and trace information encoded in the Penn-II treebank trees to associate tree nodes with f-structure equations from which a constraint solver generates f-structures. The annotation is evaluated quantitatively (Table 1) for

the 48,424 treebank trees (without FRAG or X constituents) and qualitatively (Table 2)

# f-str. frags	# sent	percent
0	120	0.25
1	48304	99.75

Table 1: Coverage & fragmentation results

	all annotations	preds-only
Precision	0.93	0.94
Recall	0.90	0.87

Table 2: Precision and Recall on f-structures against a manually encoded set of 105 gold standard f-structures from section 23

Based on this resource (Cahill *et al.*, 2002) derive two parsing architectures to parse new text: a *pipeline* and an *integrated* model. In the pipeline model, a PCFG is extracted from the unannotated treebank and used to parse new text. The resulting context-free trees are then passed into the automatic annotation algorithm to generate f-structures. In the integrated model an annotated PCFG is extracted from the f-structure-annotated Penn-II resource and used to parse new text. This results in annotated parse trees, from which an f-structure can be generated. The parsers are trained on sections 02-21 and evaluated against section 23. The grammars are both robust and wide-coverage. They achieve 81.2% f-score against the approx. 2,400 trees (section 23) and 60.6% f-score against the gold-standard f-structures. More recent experiments, with an improved annotation algorithm, yield an f-score of over 75% against the gold-standard f-structures.

4 From TIGER to a German LFG

The TIGER treebank (Brants *et al.*, 2002) is a corpus of (currently) 36,000 syntactically annotated German newspaper sentences. The annotation consists of generalised graphs, which may contain crossing and secondary edges. Edges are labelled, so that a TIGER tree encodes both phrase-structural information and dependency relations.

(Forst, 2003) converts the TIGER graphs directly into f-structures. However, in order to be able to extract an annotated PCFG which can be used to parse text into f-structures, we require trees that have been

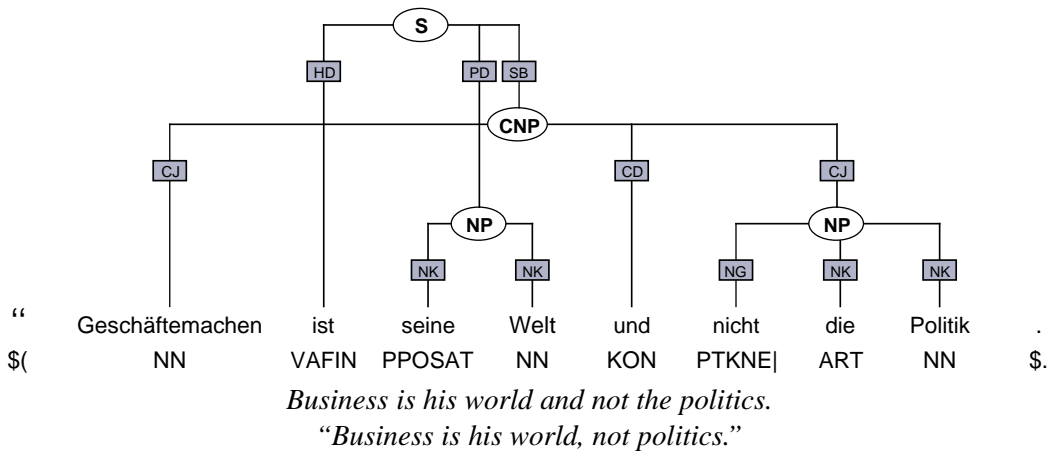


Figure 2: TIGER graph #45, containing crossing edges

annotated with f-structure equations, rather than the f-structures themselves.

Since the structure of the TIGER corpus is quite different to that of the Penn-II Treebank, the approach taken to annotating the TIGER corpus with f-structure equations differs from the approach described earlier. German does not usually rely on positional information to express functional information, a feature of English that was heavily exploited previously. However, the TIGER corpus aims to provide functional information by way of labelled edges in the graphs. By exploiting these labels we can annotate the TIGER corpus with f-structure equations.

4.1 From Graphs to Trees

The first stage in annotating the TIGER corpus with f-structure equations is to convert the TIGER graphs into trees similar to those found in the Penn-II Treebank. Traces are used to represent the crossing edges. Secondary edges have not been incorporated into the annotation procedure at this stage.² Although these edges obviously contain vital information for the generation of f-structures, this information is currently not utilised, since their treatment is unclear. However, we hope to exploit them in future work.

Figures 2 and 3 illustrate how traces are used to represent crossing edges. The TIGER graph indicates by means of crossing edges the fact that both

²Secondary edges contain information relating to re-entrancies (e.g. a shared subject in a coordination construction).

```
(
($*LRB* ``)
(S
(CNP-SB
(NN-CJ Geschäftemachen)
(*T1*-CD -)
(*T2*-CJ -)
)
(VAFIN-HD ist)
(NP-PD
(PPOSAT-NK seine)
(NN-NK Welt)
)
(KON-*T1* und)
(NP-*T2*
(PTKNEG-NG nicht)
(ART-NK die)
(NN-NK Politik)
)
)
)
($. .)
)
“Geschäftemachen ist seine Welt und nicht die Politik.”
```

Figure 3: TIGER graph #45 transformed into a Penn-II style tree with indexation

Geschäftemachen and *und nicht die Politik* form a discontinuous constituent, in the middle of which the rest of the sentence appears.³

4.2 Annotation of Derived Trees

The annotation of the trees which result from the above transformation is a two-stage process, with

³Thanks to Michael Schiehlen who provided the code to convert the graphs into corresponding trees with traces

```

(TOP
 ($*LRB* ``)
 (S [up=down]
  (CNP-SB [up-subj=down]
   (NN-CJ [down-elem=up:conj]
    Geschäftemachen)
   (*T1*-CD -)
   (*T2*-CJ -)
  )
  (VAFIN-HD [up=down] ist)
  (NP-PD [up-xcomp_pred=down]
   (PPOSAT-NK [up-spec:poss=down] seine)
   (NN-NK [up=down] Welt)
  )
  (KON-*T1* [up:subj=down] und)
  (NP-*T2* [down-elem=up:subj:conj]
   (PTKNEG-NG [down-elem=up:adjunct]
    nicht)
   (ART-NK [up-spec:det=down] die)
   (NN-NK [up=down] Politik)
  )
 )
 )
 ($ . .)
 )

```

Figure 4: The tree in Figure 3 after automatic annotation

both pre- and post-editing of the annotated trees.

The preprocessing is a simple walk through the tree in order to build a lookup table for the trace nodes. This is needed since often the trace occurs before the actual node in the tree and the information on the actual node is needed in order to assign an f-structure equation to the trace node.

The first stage attempts to assign an f-structure equation to each node based on the functional labels in the tree. We have compiled a lookup table with default f-structure equations for each functional label. E.g., the default entry for the SB (subject) label is $\uparrow \text{SUBJ} = \downarrow$. It is also possible to overwrite the default entries. E.g. the NK label (noun kernel element) alone is often ambiguous, though given some context, it is often straightforward to determine the f-structure equation required. For example an ART (article) node with an NK label can usually be annotated $\uparrow \text{SPEC:DET} = \downarrow$.

The second stage in the annotation process involves overwriting the default annotations in certain situations. These include:

- Determining the object of pre- and post-positions, labelled AC (adpositional case

marker);

- Determining the behaviour of the CP (complementiser) labelled node;⁴
- Determining the head of a coordination phrase with more than one coordinating conjunction.

Figure 5 illustrates how the flat analysis of a German PP can be annotated to give the correct f-structure analysis.

Finally, a post-processing stage explicitly links trace nodes and the reference node. This involves adding equations such as $\uparrow \text{XCOMP:OBJ} = \downarrow$ to nodes with trace information.

Figures 3 and 4 illustrate a complete annotation of a TIGER tree and Figure 6 illustrates the resulting f-structure.

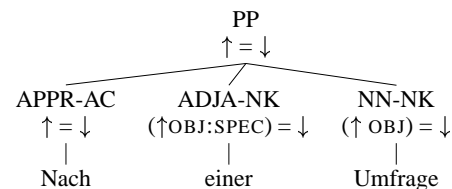


Figure 5: A flat analysis of a German PP and its f-structure annotations

4.3 Extracting an LFG from TIGER

Using the above annotation method we automatically annotate the TIGER corpus with f-structure equations. We can then quite simply read off a CFG from this annotated corpus, resulting in an *annotated* PCFG for German. We then use a standard parser to parse with this grammar, using Viterbi pruning to always obtain only the most probable parse. Using the same method as described in (Cahill *et al.*, 2002) we collect the f-structure annotations from the resulting parse tree and use a constraint solver to produce an f-structure.

5 Evaluation

In order to evaluate the quality of the grammar extracted, we set sentences 8001-10000 aside for

⁴In our analysis, true complementisers, i.e. *daß* and *ob*, only contribute a COMP-FORM feature to the f-structure, whereas other conjunctions contribute a semantic form that governs an object.

```

subj : conj : 1 : pred : 'Geschäftemachen'
      2 : spec : det : pred : die
      adjunct : 3 : pred : nicht
                pred : 'Politik'
      coord_form : und
xcomp_pred : spec : poss : pred : pro
              pred : 'Welt'
pred : ist

```

Figure 6: The f-structure produced as a result of automatically annotating the tree in Figure 3

testing purposes and developed a gold standard of 100 sentences extracted randomly from these sentences. The 100 sentences were first converted to f-structures using the methodology outlined in (Forst, 2003). They were then converted into a set of dependency relations, similar to those of the PARC Dependency Bank (King *et al.*, 2003), which were then checked and corrected manually. Using this gold-standard it was possible to use various evaluation metrics to evaluate the grammar.

5.1 Evaluation of the Automatic Annotation Algorithm

We first established the coverage of the annotation algorithm on the entire TIGER corpus. Table 3 illustrates the results. 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 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 if which are caused by annotation discrepancies. We also evaluate the quality of the annotation against our manually constructed gold-standard of 100 sentences. Table 4 illustrates that currently our automatic annotation receives an f-score of 85.74% when we compare the dependency relations generated by automatically annotating the 100 TIGER trees to the gold-standard relations. We expect this figure to improve as we refine the algorithm.

# f-str. frags	# sent	percent
0	153	0.42
1	35191	96.46
2	1054	2.89
3	77	0.21
4	2	0.006
5	1	0.003
6	1	0.003
7	3	0.008

Table 3: Coverage & fragmentation results of German Annotation Algorithm

	Preds Only Evaluation
Precision	86.79
Recall	84.71
F-Score	85.74

Table 4: Evaluation of the f-structures produced by automatically annotating the TIGER trees

5.2 Evaluation of the Grammar

An annotated grammar was extracted from the TIGER corpus (excluding the 2000 sentences set aside for testing). Using this grammar, the 2000 test sentences were parsed using Helmut Schmid’s Bit-Par parser (p.c.), which always produced the most probable analysis. 1992 sentences received a parse, from which 1974 received at least one f-structure fragment. Only 3.3% of the sentences received a fragmented f-structure, mostly due to the nature of these sentences. Table 5 contains the entire fragmentation percentages of the f-structures generated by the grammar. Table 6 illustrates the results of evaluating the trees produced by the parser against the same trees as produced in the conversion from TIGER trees to Penn-II style trees. These annotated trees were then processed into f-structures. To evaluate the quality of the f-structures produced by the parser, we again evaluated the 100 sentences selected randomly from the 2000 test sentences. Table 7 illustrates the results.

# f-str. frags	# sent	percent
0	26	1.3
1	1908	95.4
2	61	3.05
3	5	0.25

Table 5: Coverage & fragmentation results of parsing with the annotated grammar

	Unlabelled	Labelled
Precision	66.05	63.22
Recall	69.29	66.33
F-Score	67.63	64.74

Table 6: Evaluation of the trees produced by the parser

	Preds Only Evaluation
Precision	70.17
Recall	62.70
F-Score	66.23

Table 7: Evaluation of the f-structures produced by the parser

6 Lexical Resources and LDDs

In ongoing work we have extracted lexical resources such as subcat frames (LFG semantic forms) and long-distance dependency paths from the f-structures generated from the Penn-II treebank. (Cahill *et al.*, 2003b) We apply these resources during parsing to resolve long-distance dependencies in f-structure. (Cahill *et al.*, 2003a). We expect to be able to extract similar resources from the f-structure annotated TIGER bank.

7 Conclusions

We have outlined what we believe is a novel, semi-automatic approach to multilingual grammar development based on treebank resources and automatic f-structure annotation algorithms. This method can offer substantially reduced grammar development cost if a treebank is available. Depending on the size of the treebank the method can deliver robust and wide-coverage unification grammars. The method has been developed and tested for English (Cahill *et al.*, 2002). We have illustrated how this method can be adapted to German and the TIGER treebank resource. We present results of parsing 2000 sentences with a probabilistic LFG extracted automatically from the f-structure annotated TIGER treebank. The parser produces f-structures which receive an f-score of 66.23 when evaluated against our gold-standard. We have argued for LFG as a suitable framework for multilingual grammar development. However, nothing in the methodology pre-

sented here precludes their application to other languages or corpora. In addition, different annotation schemes could also be applied to automatically derive other representations, e.g. HPSG attribute-value structures, dependency structures etc. Finally, we do not believe that semi-automatic, treebank-based, multilingual grammar development necessarily competes with traditional, predominantly manual grammar development: indeed, flexible integration of hand-crafted, deep, HPSG grammars and automatically extracted, pure CFG-based topological treebank grammars (not unification grammars) has recently been demonstrated in (Crysmann *et al.*, 2002; Frank *et al.*, 2003b) and we consider this to be a promising direction for future research in both mono- and multilingual wide-coverage, deep unification grammar development.

References

- Becker, M. and A. Frank 2002. ‘A Stochastic Topological Parser for German’. *Proceedings of COLING 2002*, Taipei, Taiwan.
- 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
- Brants, T. and O. Plaehn 2000. ‘Interactive Corpus Annotation.’ In *Second International Conference on Language Resources and Evaluation (LREC-2000)*, Las Palmas, the Canary Islands.
- Bresnan, J. 2001. *Lexical-Functional Syntax*. Blackwell, Oxford.
- Butt, M., T.H. King, M.E. Niño and F. Segond, 1999. *A grammar writer’s cookbook*. Stanford, CA. CSLI Publications
- 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, R. O’ Donovan, J. van Genabith and A. Way; 2003. ‘Lexicalisation of Long-Distance Dependencies in a Treebank-Based, Statistical LFG Grammar.’ In: *Proceedings of the Eighth*

- International Conference on LFG* Albany, NY. (to appear)
- Caill, A., M. McCarthy, R. O' Donovan, J. van Genabith and A. Way; 2003. 'Extracting Large-Scale Lexical Resources for LFG from the Penn-II Treebank.' In: *Proceedings of the Eighth International Conference on LFG* Albany, NY. (to appear)
- Crysmann, B., A. Frank, B. Kiefer, St. Müller, G. Neumann, J. Piskorski, U. Schäfer, M. Siegel, H. Uszkoreit, F. Xu, M. Becker and H. Krieger 2002. 'An Integrated Architecture for Deep and Shallow Processing' *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)* Philadelphia, PA.
- Forst, M. 2003. 'Treebank Conversion - establishing a test suite for a broad-coverage LFG from the TIGER treebank' In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora (LINC'03)*, Budapest, Hungary
- Frank, A. 2000. 'Automatic F-Structure Annotation of Treebank Trees.' In: (eds.) M. Butt and T. H. King, *The fifth International Conference on Lexical-Functional Grammar*, The University of California at Berkeley, 19 July - 20 July 2000, CSLI Publications, Stanford, CA.
- Frank, A., L. Sadler, J. van Genabith and A. Way 2003. 'From Treebank Resources to LFG F-Structures.' In: (ed.) Anne Abeille, *Treebanks: Building and Using Syntactically Annotated Corpora*, Kluwer Academic Publishers, Dordrecht/Boston/London, to appear (2003)
- Frank, A., M. Becker, B. Crysmann, U. Kiefer and U. Schaefer 2003. 'TopP meets HPSG' *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)* Sapporo, Japan. (to appear)
- Gamon, M., C. Lozano, J. Pinkham and T. Reutter 1997. 'Practical Experience with Grammar Sharing in Multilingual NLP.' In J. Burstein and C. Leacock (eds.): *Proceedings of the workshop From Research to Commercial Applications: Making NLP Work in Practice*, at the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL'97), July 11-12, 1997. Madrid, Spain. Madrid: UNED, 49-56.
- Kaplan, R. and J. Bresnan 1982. 'Lexical-functional grammar: a formal system for grammatical representation.' In Bresnan, J. (ed.) *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge Mass. 173-281.
- King, T.H., R. Crouch, S. Riezler, M. Dalrymple, and R.M. Kaplan 2003. 'The PARC 700 Dependency Bank.' In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora*, EACL'03, Budapest.
- 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*. Princeton, NJ.
- Riezler, S., T.H. King, R. 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.
- van Genabith, J. and R. Crouch 1996. 'Direct and Underspecified Interpretations of LFG f-Structures.' In: *COLING 96*, Copenhagen, Denmark, *Proceedings of the Conference*. 262-267.