# Treebank Conversion
## Creating a German f-structure bank from the TIGER Corpus

Martin Forst

University of Stuttgart
Institute for Natural Language Processing
forst@ims.uni-stuttgart.de

**Proceedings of the LFG03 Conference**

University of Albany, State University of New York

Miriam Butt and Tracy Holloway King (Editors)

2003
CSLI Publications
http://csli-publications.stanford.edu/

**Abstract**

This paper reports on the conversion of the TIGER treebank, a syntactically interpreted corpus of German newspaper texts, into a testsuite for a broad-coverage Lexical-Functional Grammar (LFG) for German. It presents the two major steps of the conversion, which consists of an XSLT transformation of the TIGER XML representation into a relational Prolog-like representation and the subsequent application of term-rewriting rules as they are used in certain MT transfer components to that representation. Then some problems due to considerable differences in analysis or to information not encoded in the TIGER representation are discussed. The output consists of (partly ambiguous) f-structure charts, which can then be mapped against the grammar's output for evaluation purposes.

## 1 Introduction

In grammar development, the lack of large annotated testsuites is a serious obstacle to the further extension and adaptation of the grammars concerned, because it makes it extremely costly to evaluate grammars systematically and to keep track of the consequences of grammar modifications on coverage, efficiency and accuracy. At present, without large testsuites for German, a linguist involved in the development of a broad-coverage grammar such as the German ParGram LFG can of course run the grammar on large corpora and state afterwards how many sentences in a given corpus were parsed, what percentage timed out or failed because of storage overflow, and how many did not get any analysis. It is virtually impossible, however, to determine the accuracy of the analyses obtained, which relativizes the informational value of the given figures considerably.

Moreover, large f-structure banks lend themselves to the (supervised) training of probabilistic disambiguation modules, which select the most probable parse out of the sometimes extremely numerous analyses proposed by the symbolic grammar. (See Riezler et al. (2002) for the development of such a module for the English ParGram LFG.) It is thus evident that the creation of such f-structure banks would be a considerable advance in LFG grammar engineering.

Since the manual annotation of such testsuites would be extremely time-consuming, it seems reasonable to use an existing treebank, the TIGER corpus in our case, and to convert it to the format we need, which is the one of LFG f-structures.

Similar efforts of f-structure annotation of treebanks have been reported on in Van Genabith et al. (1999), Sadler et al. (2000), Frank (2000), Frank et al. (2001), Van Genabith et al. (2001), and Cahill et al. (2002). Since in all that work the source format (AP corpus, Susanne corpus, Penn treebank) differs considerably from the TIGER format in that it encodes mainly phrase-structural information, our approach is quite different, however, from the ones mentioned. Dependency information being expressed explicitly in the edge labels, we do not need to f-annotate the treebank (or a context-free grammar extracted from it), but we can directly convert the hybrid TIGER representation into f-structures.

Another related work is Frank (2001), which consists of the extraction of an LTAG from the NEGRA corpus. Here, the source format is comparable to ours, the TIGER format being an extension of the NEGRA format, and the main differences with respect to our work are due to the different target format. For the conversion of the corpus to a collection of f-structures, constituency information is almost irrelevant, whereas it is crucial for the extraction of an LTAG.

Finally, our conversion is, of course, in many ways similar to the inverse conversion from LFG analyses to TIGER trees Zinsmeister et al. (2002). E.g. we use the same term-rewriting system. However, since the relation between TIGER trees and f-structures is far from being a one-to-one mapping, it raises new questions. Moreover, we aim at converting the entire TIGER treebank into an 'f-structure bank' with hardly any human intervention, an objective that is quite different from grammar-based treebank annotation.

Our presentation of the conversion process is organized as follows. Section 2 describes the first step in the process, which is the transformation of TIGER trees into feature structures. In section 3 we present the transfer system we use for the transformation of TIGER-like feature structures into f-structures, as well as a number of transfer phenoma and their treatment in that formalism. Section 4 presents some results. Finally, section 5 gives an outlook on the possibilities for future work offered by the resulting f-structure bank.

## 2 The TIGER treebank and the relational TIGER representation

The TIGER Corpus comprises currently 40,000 syntactically annotated German newspaper sentences. The annotation consists of generalized graphs, i.e. trees which may contain crossing and secondary edges. Edges are labeled, so that a TIGER tree encodes both phrase-structural information and information on dependency relations.[1]

The TIGER trees are represented in a specific XML format, the so-called TIGER XML.[2] Figure 2 illustrates what the TIGER XML representation of an annotated sentence like the one in figure 1 looks like.

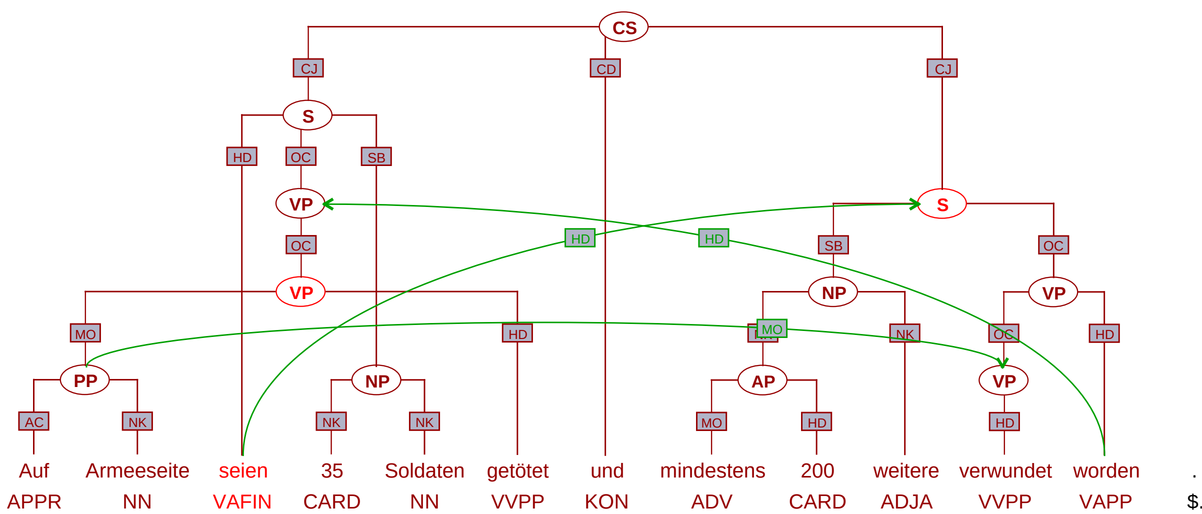


Figure 1: TIGER tree representation of corpus sentence no. 1376

In order to be able to use the XTE term-rewriting system for the conversion of the TIGER trees into f-structures, we first need to have the TIGER corpus available in a relational Prolog-like representation. Instead of being a generalized graph, a TIGER tree then has to take the form of a feature structure.

This conversion raises a first problem: In a TIGER tree, there can be several identically labelled edges that go from one single node to various of its daughter nodes. In feature structures, on the contrary, a given attribute can only have one unique value. It is thus not possible to convert a TIGER tree into a feature structure by a one-to-one mapping. Fortunately, there is quite a straightforward solution to this problem: As attributes in a feature structure can be set-valued, all identically labeled daughter nodes of a given node can be put into a set. The resulting representation differs somewhat from the initial tree, but it contains basically the same information.

[1]For more details on the annotation scheme see Skut et al. (1997), Brants & Hansen (2002), and Brants et al. (2002).
[2]See Mengel & Lezius (2000).

```
...
 <t id="s1376_7" word="und" pos="KON" morph="--"/t>
 <t id="s1376_8" word="mindestens" pos="ADV" morph="--"/t>
 <t id="s1376_9" word="200" pos="CARD" morph="--"/t>
 <t id="s1376_10" word="weitere" pos="ADJA" morph="--"/t>
 <t id="s1376_11" word="verwundet" pos="VVPP" morph="--"/t>
 <t id="s1376_12" word="worden" pos="VAPP" morph="--">
  <secedge label="HD" idref="s1376_507" />
 </t>
 <t id="s1376_13" word="." pos="$." morph="--"/t>
</terminals>
<nonterminals>
 <nt id="s1376_500" cat="PP">
  <edge label="AC" idref="s1376_1" />
  <edge label="NK" idref="s1376_2" />
  <secedge label="MO" idref="s1376_503" />
 </nt>
 <nt id="s1376_501" cat="NP">
  <edge label="NK" idref="s1376_4" />
  <edge label="NK" idref="s1376_5" />
 </nt>
 <nt id="s1376_502" cat="AP">
  <edge label="MO" idref="s1376_8" />
  <edge label="HD" idref="s1376_9" />
 </nt>
...
```

Figure 2: excerpt of the TIGER XML representation of corpus sentence no. 1376

Another problem that we need to deal with when converting TIGER trees into feature structures is the fact that, generally, the latter do not encode any information about precedence relations. This kind of information can be crucial, however, for subsequent steps in the conversion from one format to the other. Genitive attributes, for example, are labeled AG in the TIGER treebank, whether they are on the left or on the right of their head noun. The broad-coverage LFG for German, on the contrary, analyzes them in two different ways, either as a SPEC POSS, when they are in prenominal position, or as a member of the set-valued feature ADJUNCT, when they appear postnominally. This means that a minimum of information about precedence needs to be encoded in the relational TIGER representation.

This can be done with the help of a the special XLE predicate 'scopes' that allows us to state that a certain node A precedes another node B. By means of 'scopes', we express precedence relations between daughters of the same mother node. This kind of information is sufficient to disambiguate all TIGER-LFG mismatches which can be disambiguated on the basis of precedence information.

The first step of the conversion of TIGER trees into f-structures thus consists of transforming the trees into feature structures. As this task does not require any major structural changes, it can be carried out quite comfortably by means of an XSL style sheet[3]. Figure 3 shows an excerpt of the relational Prolog-like representation of the corpus sentence displayed in figure 1 that results from the XSL conversion of the TIGER XML representation illustrated in figure 2. Figure 4 displays the corresponding feature structure.

## 3 Treebank conversion by (MT) transfer rules

Although the f-structures we obtain from our broad-coverage LFG and the TIGER treebank representations coincide in core aspects, e.g. the encoding of grammatical functions, there are mismatches in

[3]Thanks to Hannes Biesinger for a first version of the XSL style sheet and to Stefanie Dipper for her contribution to its final adaption.

```
...
, cf(1, eq(attr(var(6), 'TI-FORM' ), 'getötet'))
, cf(1, eq(attr(var(6), 'TI-ID' ), 6))
, cf(1, eq(attr(var(6), 'TI-POS' ), 'VVPP'))
, cf(1, eq(attr(var(504), 'MO' ), var(1011504)))
, cf(1, in_set(var(500), var(1011504)))
, cf(1, eq(attr(var(500), 'TI-CAT' ), 'PP'))
, cf(1, eq(attr(var(500), 'TI-ID' ), 500))
, cf(1, scopes(var(1), var(2)))
, cf(1, eq(attr(var(500), 'AC' ), var(1001500)))
, cf(1, in_set(var(1), var(1001500)))
, cf(1, eq(attr(var(1), 'TI-FORM' ), 'Auf'))
, cf(1, eq(attr(var(1), 'TI-ID' ), 1))
, cf(1, eq(attr(var(1), 'TI-POS' ), 'APPR'))
, cf(1, eq(attr(var(500), 'NK' ), var(1012500)))
, cf(1, in_set(var(2), var(1012500)))
, cf(1, eq(attr(var(2), 'TI-FORM' ), 'Armeeseite'))
...
```

Figure 3: excerpt of the relational Prolog-like representation of corpus sentence no. 23474

analysis details that are comparable to translation mismatches in natural language translation. One such phenomenon is the flat analysis of auxiliary constructions generally adopted in LFG versus the intricate analysis that has been chosen for the TIGER treebank. This kind of mismatches motivates the use of transfer technology originally developed for machine translation.

### 3.1 The transfer system

The transfer system we use is a term rewriting system based on Prolog. It has originally been developed by Martin Kay and is now part of the XLE grammar development platform. [4] The rules it processes are ordered, which means that the output of a given rule $r_i$ is input to rule $r_{i+1}$. Each rule replaces a certain set of predicates (those on the left-hand side of the rule) by another set of predicates (those on its right-hand side). Input and output predicates are separated by a rewriting symbol, the operator '==>'. The most basic rules simply rewrite the name of the predicate and pass on the values of the arguments unchanged. For example, the rule given in (1) maps the TIGER edge label OA (accusative object) to the LFG function OBJ.

(1) `oa(X,Y) ==> obj(X,Y).`

In addition, it is possible to specify predicates on the left-hand side that have to be matched, but are not replaced (marked with a '+'), as well as predicates that must not be matched for the rule to be applied (marked with a '-'). These mechanisms are used in the following rule, which takes a partial feature structure whose attribute TI-POS has the value PIAT (for 'attributive indefinite pronoun') out of the set that is the value of the feature NK (for 'noun kernel') and attributes it to a new feature SPEC QUANT, if that feature does not yet exist.

(2)
```
+nk(A,SET), in_set(B,SET), +ti_pos(B,'PIAT'), -spec(A,_) ==>
spec(A,SPEC), quant(SPEC,B).
```

It is also possible to delete features by writing a zero on the right-hand side of a rule, which stands for the empty set. In this case, all predicates on the left-hand side of the rule are deleted from the set of terms without replacement.

[4] I would like to thank Anette Frank (DFKI Saarbrücken) for her input and great help with the transfer component.

```
"Auf Armeeseite seien 35 Soldaten getötet und mindestens 200 weitere verwundet worden . "

[CD {-11[TI-FORM und, TI-ID 7, TI-LEMMA und, TI-POS KON]}
 CJ { 
      [HD [TI-FORM seien, TI-ID 3, TI-LEMMA sein, TI-POS VAFIN]
       OC [HD [TI-FORM worden, TI-ID 12, TI-LEMMA werden, TI-POS VAPP]
           OC [HD [TI-FORM verwundet, TI-ID 11, TI-LEMMA verwunden, TI-POS VVPP]
               MO [AC {-3[TI-FORM Auf, TI-ID 1, TI-LEMMA auf, TI-POS APPR]}
                   NK {   [MOD {-1[TI-LEMMA Armee]}
                       -2  TI-FORM Armeeseite, TI-ID 2, TI-LEMMA Seite, TI-POS NN]}
                   TI-CAT PP, TI-ID 500]
               TI-CAT VP, TI-ID 503]
           TI-CAT VP, TI-ID 506]
       SB [NK {-8[DEGREE comparative_, TI-FORM weitere, TI-ID 10, TI-LEMMA weit, TI-POS ADJA]
               [HD [TI-FORM 200, TI-ID 9, TI-LEMMA 200, TI-POS CARD]
                MO {-6[TI-FORM mindestens, TI-ID 8, TI-LEMMA mindestens, TI-POS ADV]}
             -7 TI-CAT AP, TI-ID 502]}
           TI-CAT NP, TI-ID 505]
   -10 TI-CAT S, TI-ID 508]
      [HD [-10-HD]
       OC [HD [-10-OC-HD]
           OC [HD [TI-FORM getötet, TI-ID 6, TI-LEMMA töten, TI-POS VVPP]
               MO <[-10-OC-OC-MO]>
               TI-CAT VP, TI-ID 504]
           TI-CAT VP, TI-ID 507]
       SB [NK {-5[TI-FORM 35, TI-ID 4, TI-LEMMA 35, TI-POS CARD]
               -4[TI-FORM Soldaten, TI-ID 5, TI-LEMMA Soldat, TI-POS NN]}
           TI-CAT NP, TI-ID 501]
    -9 TI-CAT S, TI-ID 509]
    }
-12 TI-CAT CS, TI-ID 0]
```

Figure 4: representation of corpus sentence no. 1376 as a TIGER-annotated feature structure

(3) `ti_form(_,_) ==> 0.`

Finally, the possibility of defining rules as optional needs to be mentioned as well. Optional rules are characterized by the use of the operator '?=>' instead of '==>'. They allow us to transfer a given input feature structure to two alternative output structures - or more, if several optional rules are applied. We can thus handle cases where we cannot clearly decide on the sole basis of the input what the output must look like. The TIGER label MO (for 'modifier'), for example, is such a phenomenon, because the context is not always sufficient to determine whether it is to be transferred to an element of the set-valued feature ADJUNCT, to an OBL-DIR (directional oblique), an OBL-LOC (locative oblique) or still another grammatical function. The following rule optionnally transfers a MO-PP with an AC (the edge label used for pre- and postpositions in TIGER) that has the form 'nach' into an OBL-DIR.

(4)
```
+ti_cat(S,'S'), +mo(S,MO), in_set(PP,MO), +ti_cat(PP,'PP'),
+ac(PP,APPR), +ti_form(APPR,'nach') ?=>
obl_dir(S,PP).
```

For reasons of userfriendliness and maintainability, the XLE transfer system also allows the use of templates and macros. They are short-hand notations for sets of rules and predicates respectively. As they are not directly relevant for our presentation, however, we do not present them here in more detail.

## 3.2 Transfer phenomena

Unlike transfer in machine translation, the transfer from TIGER trees to LFG f-structures does not aim at changing the surface string. The task is rather to map a limited set of grammatical features into another limited set of grammatical features. Nevertheless, the format conversion is far more complex than a simple mapping from one feature set to another, because (i) there is no one-to-one correspondence between features, (ii) the different analyses chosen for certain grammatical phenomena can have relatively heavy repercussions on the structure of the representations involved, and (iii) the TIGER Corpus graphs contain most, but not all information needed for the conversion to f-structures.

### 3.2.1 Ambiguous edge labels

In section 2, we mentioned the case of the TIGER edge label AG, which depending on the position of the AG constituent with respect to its head noun corresponds to either a SPEC POSS feature or an ADJUNCT feature in a German LFG analysis. Still, this kind of ambiguity can easily be resolved on the basis of precedence information, so that we simply need two obligatory rules for the transfer of AGs, one for prenominal ones and a 'default rule' for postnominal ones. As rules are ordered, the 'default rule' is only applied, if the more specific rule was not.

(5) a.
```
+ti_cat(NP,'NP'), +nk(NP,NKSET), +in_set(HEAD,NKSET),
+ti_pos(HEAD,'NN'), +scopes(AG,HEAD), ag(NP,AG) ==>
spec(NP,SPEC), poss(SPEC,AG).
```
b.
```
ag(NP,AG) ==>
adjunct(NP,ADJUNCT), in_set(AG,ADJUNCT).
```

A somewhat more complex case is the transfer of the predicate MO. It can correspond to the predicates ADJUNCT, OBL-DIR and OBL-LOC. This is due to the fact that PPs such as *auf Armeeseite* in corpus sentence no. 1376 are always annotated as MOs in the TIGER Corpus, whereas the German LFG analyses them as subcategorized arguments in some contexts.

We deal with this case by first using the optional rule in (6a), which similarly to the one in (4) converts a MO into an OBL-LOC, and then applying the default rule given in (6b), which transfers all MOs to ADJUNCTs. In order not to obtain too many output f-structures, we try to limit the application of the optional rules to as few contexts as is reasonably possible, while keeping them general enough to cover all cases that we need for a justifiable comparison of the output of the German LFG and the TIGER annotation. Nevertheless, the optional rules we use give rise to a considerable amount of ambiguity, which can easily been seen in figure 5.

(6) a.
```
+ti_cat(S,'S'), +mo(S,MO), in_set(PP,MO), +ti_cat(PP,'PP'),
+ac(PP,APPR), +ti_form(APPR,'auf') ?=>
obl_loc(S,PP).
```
b.
```
mo(S,MO) ==> adjunct(S,MO).
```

"Translation of: Auf Armeeseite seien 35 Soldaten getÃ¶tet und mindestens 200 weitere verwundet worden

```
    PRED      'verwunden'
              PRED     'pro'
                         PRED    'weit'
              ADJUNCT  { SUBJ    [PRED      'pro']
                                 [PRON-TYPE null ]
    SUBJ               -3 DEGREE comparative }
              SPEC     NUMBER [PRED     '200'
                               ADJUNCT {-2[PRED 'mindestens']}]
                   PRED 'auf'
    ADJUNCT  {     OBJ  [PRED 'Seite'
               -4        MOD  {-1[PRED 'Armee']}] }
    TNS-ASP  [ASPECT   [PERF +]
 -6           PASS-SEM dynamic]
    PRED      'tÃ¶ten'
    SUBJ      [PRED 'Soldat'
               SPEC [NUMBER [PRED '35']]]
    OBL-DIR   [=(<b:1 [-4:auf]>)]
    OBL-LOC   [=(<c:1 [-4:auf]>)]
    ADJUNCT   { c:2   [-4:auf]}
    OBL       [=(<a:1 [-4:auf]>)]
    TNS-ASP   [ASPECT   [PERF +]
 -5            PASS-SEM dynamic]
-7 COORD-FORM und
```

Figure 5: F-structure derived from the TIGER tree associated to corpus sentence no. 1376

### 3.2.2 Structural changes

Given that TIGER trees on the one hand encode information about both phrase structure and dependency relations and that f-structures on the other hand only represent the latter type of information, it is not surprising that the analysis of a few grammatical phenomena differs considerably between the TIGER corpus and the German LFG analyses. This is the case of analytic tenses, for example, which generally get a flat analysis in LFG, the auxiliary and the main verb being treated as f-structure co-heads, whereas in TIGER the VP containing the non-finite main verb form is analysed as a clausal object (OC) of the auxiliary. By comparing figure 5, which shows the f-structure derived from the TIGER tree in figure 1, and that same figure you can state that the resulting structure is less intricate.

This kind of structural change is known as head-switching in the field of machine translation. As studies about the treatment of head-switching phenomena have shown, they can be dealt with without major difficulty by a term-rewriting system.

## 3.3 Information not present in the TIGER Corpus

A more problematic aspect of the conversion of TIGER graphs to LFG f-structures is the fact that there is no information whatsoever in the annotation about the subjects of infinitives. For f-structures that respect completeness and coherence, this information is needed, however.

Our approach to solve this problem is to use the control information contained in the lexicon of our grammar. The drawback of this method is that erroneous or missing information in the lexicon will be reflected in the transfer output, but we judged it the most adequate way to assign subjects to infinitives without human intervention.

## 4 Results

An evaluation of our treebank conversion has been carried out on the basis of sentences 8001 to 8200 of the TIGER Corpus. [5] For this purpose, we established a gold standard for those sentences using roughly the same methodology and format as King et al. (2003) did for the PARC 700 Dependency Bank. Then the result of our treebank conversion was matched against this gold standard, taking into account predicate-argument relations only. (The current TIGER release is not yet annotated morphologically.) The matching was done in such a way that an f-structure chart counted as a complete match, if the correct analysis was among the readings contained in it.

Of the 200 sentences, 6 consisted of one single word, which makes it impossible to match predicate-argument relations, and 10 could not be converted to feature structures, because they were sequences of syntactic phrases rather than sentences (example: *dah FRANKFURT A. M. , 6. November* .). This left us with 184 sentences for evaluation, of which 9 didn't entirely match due to errors in the original TIGER annotation. Out of the 175 correctly annotated sentences, 1 failed to match due to an erroneous lemmatization and 13 had not been transferred correctly. 161 out of the 175 sentences were converted correctly, which corresponds to 92

## 5 Outlook to the use of the resulting 'f-structure bank' in grammar development

Having this large German f-structure bank available will make it possible to evaluate the German ParGram LFG in a much more informative way than this can be done at the moment. No longer will we be restricted to observing what percentage of a given corpus can be parsed by the grammar, what proportion fails due to timeout or storage overflow and what percentage is rejected, but we will have the means to determine whether the desired analysis is among the analyses proposed by the grammar and whether it is among the preferred solutions.[6]

In a preliminary experiment we evaluated the analyses proposed by the German ParGram LFG against the gold standard established for sentences 8001 through 8200, in order to get an idea of the quality of our analyses. Out of the 200 sentences, 151 received at least one parse. (For the moment, we do not use partial parsing techniques, although this is planned for the near future.) 4 of them could not be used for the matching of predicate-argument relations, since they consist of one single word. The evaluation of the remaining 147 sentences yielded a precision of 0.814, a recall of 0.809 and thus an f-score of 0.812 (upper-bound). By mapping against an f-structure bank derived from the whole TIGER Corpus, we will have an even more complete and detailed picture of how good (or bad) the analyses are which the German ParGram LFG proposes, and it will no longer be a problem to control the repercussions of grammar modifications intended to increase coverage or efficiency on parse quality.

Last but not least, the resulting f-structure bank will be indispensable for the supervised training of the disambiguation modules we plan to use along with the German ParGram LFG. These will be, on the

[5] Aoife Cahill (Dublin City University) and myself decided once that sentences 8001 through 10000 would serve as unseen test section for work done on the basis of the TIGER corpus. This corpus section was selected aleatorily.

[6] XLE provides a non-statistical OT-inspired disambiguation method, which prefers or disprefers certain solutions with respect to other ones. See Frank et al. (1998).

one hand, the optimality-theoretically inspired disambiguation module already implemented in XLE, for which we plan to learn the OT ranking in a stochastic OT fashion, and on the other hand discriminative estimation techniques as used by Riezler et al. (2002). It is important to consider the two techniques, because the OT-inspired module allows to cut down the number of preferred analyses in a very efficient way, whereas the log-linear models allow more fine-grained disambiguation.

In conclusion, treebanking along with treebank conversion opens up a whole series of new possibilities for the development of fine-grained syntactic analyzers. Most importantly, it will permit the use of probabilistic disambiguation based on supervised training and facilitate detailed grammar evaluation. And since it is relatively straight-forward to convert f-structures into more theory-neutral dependency triples like the ones of the PARC 700 Dependency Bank, it might even open the way for the comparison of different parsers for German.

## References

Bouma, G., G. van Noord, R. Malouf (2000). Alpino: Wide-coverage computational analysis of Dutch. In *Proceedings of Computational Linguistics in the Netherlands,* Amsterdam, Netherlands.

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

Brants, S., S. Dipper, S. Hansen, W. Lezius, and G. Smith (2002). The TIGER Treebank. In Hinrichs, E. & K. Simov (eds.), *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT 2002)*, Sozopol, Bulgaria.

Brants, S. & S. Hansen (2002). Developments in the TIGER annotation scheme and their realization in the corpus. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC '02)*. Las Palmas, Spain.

Cahill, A., M. McCarthy, J. van Genabith, A. Way (2002). Evaluating Automatic F-Structure Annotation for the Penn-II Treebank. In Hinrichs, E. & K. Simov (eds.), *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT 2002)*, Sozopol, Bulgaria.

Cahill, A., M. McCarthy, J. van Genabith, A. Way (2002). Automatic Annotation of the Penn-Treebank with LFG F-Structure Information In Lenci, A., S. Montemagni, V. Pirelli (eds.), *Proceedings of the LREC Workshop on Linguistic Knowledge Acquisition and Representation - Bootstrapping Annotated Language Data*, ELRA - European Language Resources Association, Paris, France.

Carroll, J., G. Minnen, Ted Briscoe (1999). Corpus annotation for parser evaluation. In *Proceedings of the EACL workshop on Linguistically Interpreted Corpora (LINC)*, Bergen, Norway.

Crouch, R., R. M. Kaplan, T. H. King, and S. Riezler (2002). A comparison of evaluation metrics for a broad-coverage stochastic parser. In *Proceedings of the "Beyond PARSEVAL" Workshop at the 3rd International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Spain.

Dipper, S. (2000). Grammar-based corpus annotation. In *Proceedings of the Workshop on Linguistically Interpreted Corpora*, Luxembourg.

Dipper, S. (2003). *Implementing and Documenting Large-scale Grammars - German LFG*. Ph.D. thesis, University of Stuttgart, Germany.

Emele, M., M. Dorna, A. Lüdeling, H. Zinsmeister, C. Rohrer (2000). Semantic-based Transfer. In: Wahlster, W. (ed.): *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer Verlag.

Frank, A., T. H. King, J. Kuhn, J. Maxwell (1998). Optimality theory style constraint ranking in large-scale lfg grammars. In Butt, M. & T. H. King (eds.), *Proceedings of the LFG98 Conference*, University of Queensland, Brisbane, CSLI Online Publications, Stanford, CA. `http://www-csli.stanford.edu/publications/`.

Frank, A. (2000). Automatic F-structure Annotation of Treebank Trees. In Butt, M. & T. H. King, (eds.), *Proceedings of the LFG00 Conference*, University of California at Berkeley, CSLI Online Publications, Stanford, CA. `http://www-csli.stanford.edu/publications/`.

Frank, A. (2001). Treebank Conversion - Converting the NEGRA treebank to an LTAG grammar. In *Proceedings of the Workshop on Multi-layer Corpus-based Analysis*, Iasi, Romania.

Frank, A., L. Sadler, J. van Genabith, A. Way (2001). From Treebank Resources to LFG F-Structures. Automatic F-Structure Annotation of Treebank Trees and CFGs extracted from Treebanks. To appear in: Abeillé, A. (ed.): *Treebanks. Building and using syntactically annotated corpora*, Kluwer Academic Publishers.

King, T. H., R. Crouch, S. Riezler, M. Dalrymple, R. M. Kaplan (2003). The PARC 700 Dependency Bank. In *Proceedings of the EACL workshop on Linguistically Interpreted Corpora (LINC '03)*, Budapest, Hungary.

Mengel, A. & W. Lezius (2000). An XML-based encoding format for syntactically annotated corpora. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC '00)*, Athens, Greece.

Riezler, S., D. Prescher, J. Kuhn, M. Johnson (2000). Lexicalized Stochastic Modeling of Constraint-based Grammars Using Log-linear Measures and EM Training. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, 2000*, Hong Kong.

Riezler, S., T. H. King, R. M. Kaplan, R. Crouch, J. T. Maxwell III, M. Johnson (2002). Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002*, Philadelphia.

Sadler, L., J. van Genabith, A. Way (2000). Automatic F-Structure Annotation from the AP Treebank. In Butt, M. & T. H. King, *Proceedings of the LFG00 Conference*, University of California at Berkeley, CSLI Online Publications, Stanford, CA. `http://www-csli.stanford.edu/publications/`.

Skut, W., B. Krenn, T. Brants, and H. Uszkoreit (1997). An Annotation Scheme for Free Word Order Languages. In *Proceedings of ANLP-97*.

Van Genabith, J., A. Way, L. Sadler (1999). Semi-Automatic Generation of F-Structures from Treebanks. In Butt, M. & T. H. King, *Proceedings of the LFG99 Conference*, University of Manchester, Great Britain, CSLI Online Publications, Stanford, CA. `http://www-csli.stanford.edu/publications/`.

Van Genabith, J., A. Frank, A. Way (2001). Treebank vs. Xbar-based Automatic Feature-Structure Annotation. In Butt, M. & T. H. King, *Proceedings of the LFG01 Conference*, University of Hong Kong, Hong Kong, CSLI Online Publications, Stanford, CA. `http://www-csli.stanford.edu/publications/`.

Zinsmeister, H., J. Kuhn, S. Dipper (2002). TIGER TRANSFER - Utilizing LFG Parses for Treebank Annotation. In Butt, M. & T. H. King, *Proceedings of the LFG02 Conference*, National Technical University of Athens, Greece, CSLI Online Publications, Stanford, CA. `http://www-csli.stanford.edu/publications/`.

Martin Forst
`forst@ims.uni-stuttgart.de`
IMS
University of Stuttgart
Azenbergstr.12
70174 Stuttgart
Germany