Towards a Toolkit Linking Treebanking to Grammar Development

Victoria Rosén, Koenraad de Smedt and Paul Meurer

University of Bergen and AKSIS
E-mail: {victoria,desmedt,paul.meurer}@uib.no

1 Introduction

An often discussed issue in treebanking is the relation between the treebank and a grammar that is at least descriptively adequate with respect to the corpus. On the one hand, the manual syntactic annotation of corpora is often advocated as an empirical source for grammar development, as opposed to introspection and constructed examples [10]. On the other hand the automatic syntactic annotation of corpora is fast and always consistent, but it requires a fully adequate grammar, which ideally should be based on a corpus. This seemingly vicious circle can be broken by an incremental approach which closely links grammar development and treebank construction. In this paper we present the development of a sophisticated toolkit, which is a prerequisite for efficiency in this approach.

The desirability of annotation that is as rich and 'deep' as possible has been pointed out earlier. Although multi-level treebanks have been constructed manually with the help of appropriate tools, for example the Prague Dependency Treebank [1], automatic annotation is very attractive in order to reach a large size [2] because increasing returns on investment are made when scaling up. Moreover, manual annotation is expensive and is no guarantee for correctness, since inter-annotator agreement usually does not exceed about 95% [2, 1]. We believe that the need for automatic annotation tools is even clearer when the envisaged complexity of the annotation is high [8]. For our purposes, manual annotation is in fact not feasible at all.

Our research is at this stage not so much oriented towards the construction of a particular treebank as towards the development of novel tools for incremental semi-automatic treebank construction tightly coupled with grammar development. Repeated reparsing of the corpus with revised grammars is feasible with efficient

parsers running on fast machines. However, manual disambiguation remains necessary. After all, even the best parser can only tell us what the most probable analysis is, not what the most plausible analysis is. It would be inefficient if reparsing the corpus were to require repeated manual disambiguation, so annotator choices must be recorded. It has been proposed to record disambiguation choices in terms of *discriminants*, elementary linguistic properties of a structure such as a particular word sense or a modifier attachment [3]. These properties are usually easy to identify independently of other properties, and they are persistent in the sense that they can be stored and reapplied in reparsing even with a revised grammar.

The value of discriminants for disambiguation has been recognized, and the concept has been applied in the context of treebanking [7, 9, 8]. In the current paper, we present advances in the construction of a treebanking toolkit that implements discriminants at several levels and we present improvements in its web-based interface. We will first outline our use of discriminants in the context of LFG-based parsing. Then, we will highlight some new features in our treebanking interface. Finally, we will discuss the linking of treebanking and grammar development.

2 Three Types of Discriminants

A discriminant is in general an elementary linguistic property of an analysis of a sentence that is not shared by all analyses. One of the main topics of our research is how discriminants may be defined and used in an optimal way in the context of treebanking. We have defined three main types of discriminants for LFG grammars: morphology discriminants, c-structure discriminants and f-structure discriminants [9, 8]. First we will briefly present the three discriminant types.

Lexical ambiguities are often the easiest for the disambiguator to decide on. It requires little or no knowledge of the grammar to decide on the intended reading of a lexical item. We have therefore implemented morphology discriminants. A word with the tags it receives from morphological preprocessing is a morphology discriminant. A lexically ambiguous sentence is shown in example 1.

(1) *Ta* *båt* *eller* *bil.*
take boat or car(N)/drive(V)
"Take a boat or a car." or "Take a boat or drive."

The word form *bil* is ambiguous; it may either be the indefinite singular of the noun *bil* "car" or the imperative of the verb *bile* "(to) drive, go by car". This ambiguity results in the morphology discriminants in table 1. The annotator may choose the noun reading or the verb reading in order to select the intended analysis of the sentence.

Table 1: Morphology discriminants for (1) *Ta båt eller bil.*

bil+Sg+Noun+Neut+Indef
bile+Verb+Impv

The lexical ambiguity in this sentence is correlated with a syntactic ambiguity. Therefore the annotator may also disambiguate it by choosing between different properties of constituent structures (c-structures). There are two subtypes of c-structure discriminants. A constituent discriminant is a top level bracketing of a constituent substring. A rule discriminant is a top level bracketing of a constituent substring labeled by the rule which induces that bracketing. The two c-structures for example 1 are shown in figure 1. These trees share many properties, but they are also different in a number of respects. There are therefore several c-structure discriminants. Four of these are shown in table 2. The top line in each row represents a constituent discriminant; the bracketing is shown by double vertical lines. The bottom line in each row represents a rule discriminant. Note that a rule discriminant is represented by a rule, but that this is an abbreviation for a substring with labeled bracketing. The representation 'I′ → Vfin S' is here a convenient shorthand for '[$_{I'}$ [$_{Vfin}$ ta][$_{S}$ båt eller bil]]'. Choosing any of these discriminants will fully disambiguate the sentence.

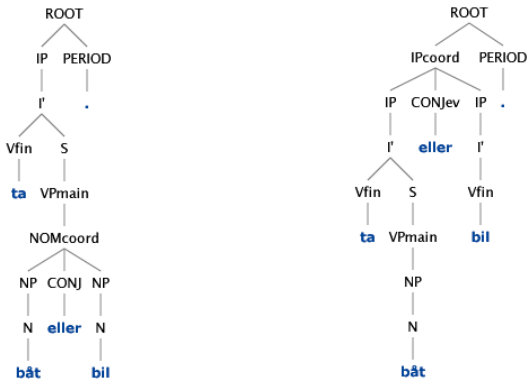

Figure 1: C-structures for (1) *Ta båt eller bil.*

Table 2: Some c-structure discriminants for (1) *Ta båt eller bil.*

ta || båt eller bil I′ → Vfin S
ta båt || eller || bil IPcoord → IP CONJev IP

It is a well-known property of Lexical-Functional Grammar that the same c-structure may project more than one well-formed f-structure. An example is provided by the sentence in 2.

(2) *Barna* *leker* *hver* *dag.*
child-DEFPL play every day
"The children play every day."

The normal interpretation of this sentence would be that the phrase *hver dag* "every day" is an adverbial. But since this phrase is an NP, it could also function as the direct object of the verb *leke* "(to) play", for example if the children had a game with this name. This is an ambiguity in the syntactic function of the NP — in LFG terms, whether it functions as OBJ or as ADJUNCT. Since there is no lexical ambiguity, there are no morphology discriminants. Neither is there any phrase structure difference between the two analyses, so that there are no c-structure discriminants. They only differ with respect to f-structure. The two f-structures for this sentence are shown in figure 2.

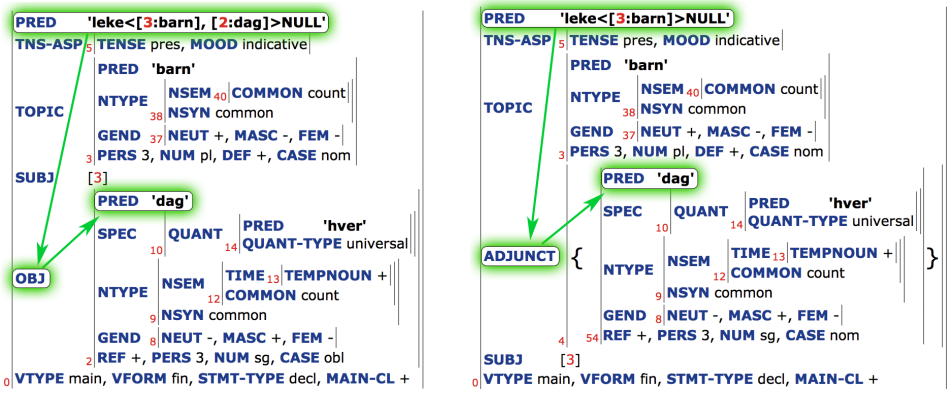

Figure 2: F-structures for (2) *Barna leker hver dag.*

These f-structures are nearly identical. The main difference between them lies in the values of the top-level PREDs. Both have the verb *leke* as predicate, but one of them is a two-place predicate and the other is a one-place predicate. The other difference is in the syntactic function assigned to the NP *hver dag*. If the main predicate is two-place, the NP functions as OBJ. If the main predicate is one-place, the NP functions as ADJUNCT. Since a sentence could have several occurrences of each of these syntactic functions, the f-structure discriminants must clearly identify

which occurrences are intended. This is done by letting f-structure discriminants be local paths through the f-structure. Two such paths have been highlighted in the f-structures in figure 2; their compact representations in the user interface are shown in table 3. The first one may be read "the two-place predicate *leke* has an OBJ whose predicate is *dag*," while the second one may be read "the one-place predicate *leke* has a set of ADJUNCTS, one of which has the predicate *dag*."

Table 3: Some f-structure discriminants for (2) *Barna leker hver dag.*

'leke<[],[]>NULL' OBJ 'dag'
'leke<[]>NULL' ADJUNCT > 'dag'

We have illustrated some simple contrasts utilizing the different types of discriminants. In reality, there are more discriminants for the given examples. But although an ambiguous sentence may have many discriminants, it is not always necessary to make a lot of choices in order to select one reading. Example 1 may be fully disambiguated by choosing one morphology discriminant or one c-structure discriminant. As part of our current research we are investigating the optimal design and use of discriminants in treebanking. The toolkit described below plays an important part in this investigation as well as being a prototype for a future annotation workbench.

3 A Web-based Treebank Toolkit

We are constructing a comprehensive treebanking toolkit based on the tight interaction between the XLE parser, a disambiguation tool, a database, the TIGERSearch tool and other supporting components. From the viewpoint of the user, the toolkit includes the following web-based interfaces:

- a sentence disambiguation page offering discriminants and a graphical display of syntactic analyses;
- an overview page supporting navigation in a chosen subcorpus as a whole;
- a discriminant statistics page that displays statistics on all chosen discriminants in a subcorpus;
- XLE-Web, an interface to the XLE parser on a web page.

These interactive components will be described in the sections below. Most of these components are implemented in Common Lisp and use XML, XSLT and Javascript to serve the interface web pages. C-structure trees (and graphs) are drawn using Scalable Vector Graphics (SVG).

3.1 The Treebank Disambiguation Interface

The annotator's main task is performed in the disambiguation interface, which presents an individual sentence from the corpus. An example for sentence 3 is given in figure 3.

(3) *Husk* *båndtvangen.*
remember leash-requirement-DEFSG
"Remember the leash law."

The annotator gets a separate web page for the disambiguation of each sentence. At the top of the interface is a roll-down menu where the annotator may choose a subcorpus to work on. Some basic numbers for the subcorpus chosen are provided: the number of sentences in the subcorpus, the number of fragment analyses, the number of sentences with no solutions, and statistics on ambiguity.

The disambiguation tool computes all the discriminants which could be used to disambiguate the sentence in question, and displays them in a table. The annotator may choose a discriminant by clicking on it, or reject a discriminant by clicking on *compl* (for complement). Each time a discriminant choice has been made, the display of discriminants is changed to those discriminants which are still relevant. Already chosen discriminants are displayed in boldface. Thus the disambiguation proceeds stepwise. The discriminating power of each discriminant is indicated as the number of remaining compatible analyses at each given moment.

In addition to examining and selecting discriminants, the annotator may choose to have packed c- and f-structures displayed [9]. Usually these structures are only of interest in disambiguation if there are not too many solutions, since when there are many solutions, the packed structures may be too complex and too large to be easily read. In fact, this is one of the main motivations for using discriminants, as we have shown [9]. As disambiguation proceeds, the packed structures become gradually less complex. When disambiguation results in one analysis, the single c- and f-structures for the sentence are displayed.

While disambiguating, the annotator may discover that the intended analysis is not present, and may also discover the reason why, for example that a certain word form does not have the necessary part of speech in the lexicon, or that a syntactic rule does not allow a necessary expansion. Such observations may be noted in the comment field and later consulted by the grammar writers during grammar and lexicon revision.

We have implemented automatic reparsing of the corpus with successive versions of the grammar, including automatic redisambiguation using the stored discriminants for each sentence. This function may be activated through the reparse button, although the normal method for reparsing is in batch mode.

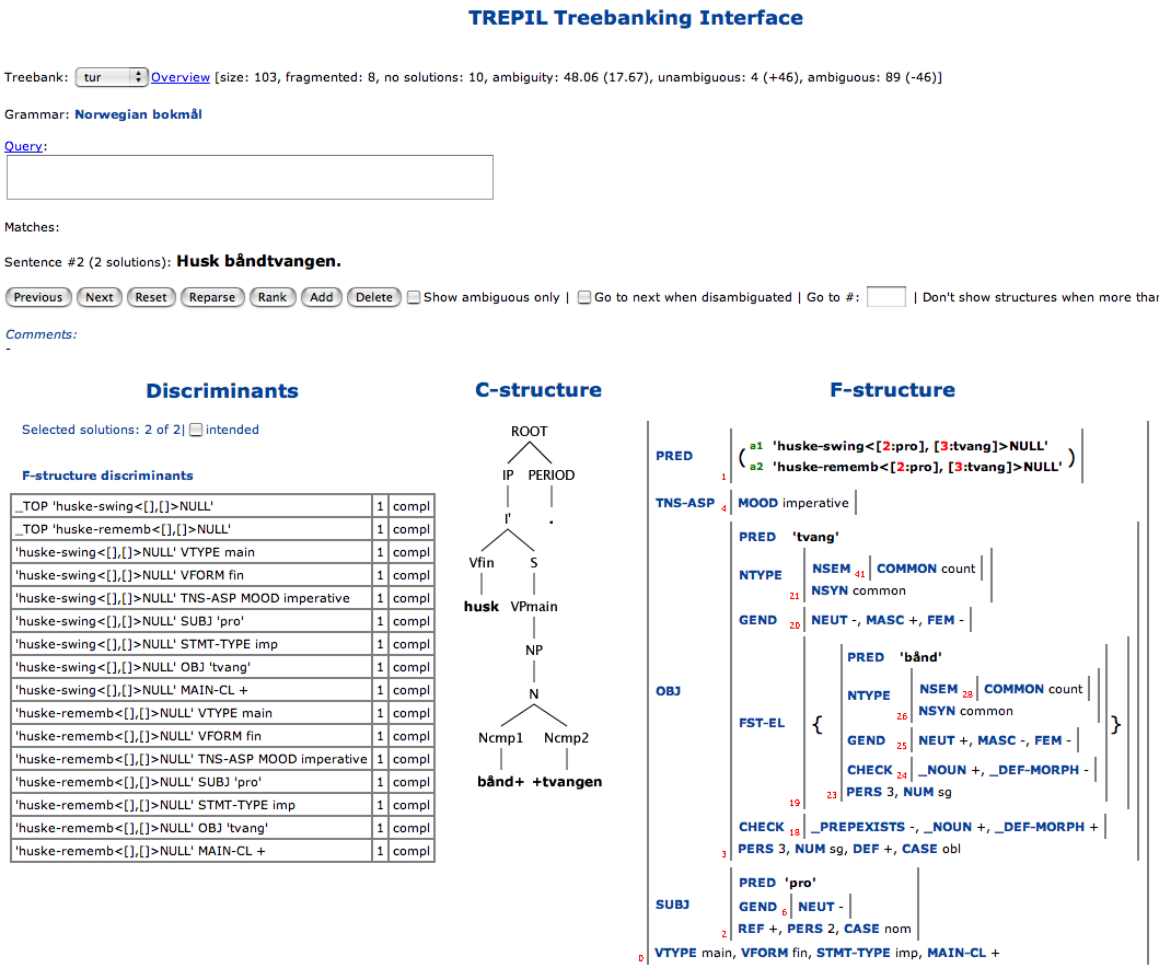

Figure 3: The TREPIL Treebank Disambiguation Tool

The annotator may navigate through the corpus in various ways. Ticking the *Show ambiguous only* and *Go to next when disambiguated* buttons will automatically bring the next ambiguous sentence up when the current sentence has been fully disambiguated. Otherwise the annotator can use the *Previous* and *Next* buttons, or write the number of a particular sentence in the *Go to #* box. Alternatively, a page with an overview of the entire subcorpus may be reached by clicking on *Overview* at the top of the page. This page is described in the next section.

3.2 The Treebank Overview and Discriminant Statistics

The purpose of the overview page is to present the results of parsing and disambiguation of the corpus so far. In figure 4 we show the top of the overview page for the subcorpus *jh2*.

Treebank: jh2
[size: 319, fragmented: 79, no solutions: 84, ambiguity: 291.91 (280.49), unambiguous: 8 (+49), ambiguous: 227 (-49)]

Grammar: **Norwegian bokmål** | Discriminant statistics

Id	*Sol.*	*Frag.*	*Disc.*	*Chosen*	*Words*	*Int.*	*Sentence*	*Comments*
12	96	*	6 of 127	**1**	15		Familien Kvame drev hotellet helt fram til 1974, da det ble solgt til Eidsbugarden Turistsenter.	This will get a full parse if '1974' is allowed to take a CPtmprel.
15	8		2 of 132	**1**	11	*	Det er merkede fotturruter til Gjendebu, Torfinnsbu, Olavsbu, Skogadalsbøen og Yksendalsbu.	
16	8		2 of 43	**1**	9	*	Vinjestova, forløperen for Eidsbugarden hotell, ble åpnet i 1868.	
17	40		3 of 116	**1**	14	*	Hotellet ligger i Vang kommune i Oppland, 1060 m o.h., og har 50 senger.	
19	**1**		0 of 0	1	12		Året etter, da DNT fylte 125 år, ble så turisthytta Fondsbu åpnet.	'året etter' should be analyzed as NP.
21	40		4 of 114	**1**	19		De 26 sengene som var i turisthytta ble raskt for få, og det ble nødvendig å bygge et anneks.	CONJspecial should not be allowed before CONJdisc.
31	**1**		0 of 0	1	13	*	Når du først er i dette området, er også Uranostind et flott turmål.	
32	2		1 of 8	**1**	5	*	Også den toppen krever brevandring.	
35	30		3 of 131	**1**	9	*	Det er bilvei til Fondsbu og båtrute over Bygdin.	
36	4		1 of 41	**1**	6	*	Fondsbu turisthytte ble åpnet i 1993.	

Figure 4: The Treebank Overview

Each overview page displays all of the sentences in the subcorpus together with various information about each sentence. The columns before the sentence show (from left to right) the identifier number of the sentence, the number of solutions, whether the sentence has only fragment analyses or full analyses, the number of discriminants chosen out of the total number of discriminants, the number of analyses chosen, the number of words in the sentence, and whether the annotator has indicated that the chosen analysis is the intended one. The contents of the page may be sorted according to any of the columns. This makes it easy for the annotator to

choose a certain category of sentences to work on if desirable, for instance short sentences, or sentences with fragment analyses. In this example, the sorting has been done for the number of chosen analyses, so that completely disambiguated sentences are shown first.

These numbers are not only useful for assessing the current state of the corpus, but they also shed light on the usefulness of the disambiguation strategy. The numbers in figure 4 demonstrate that few discriminants need to be chosen, both with respect to the total number of solutions and the total number of discriminants, in order to fully disambiguate a sentence.

In fact, we analyzed the figures for the first 101 syntactically ambiguous sentences from a subcorpus that were fully disambiguated with our toolkit. They show that the average number of chosen discriminants per sentence was as low as 2.6 and the largest number was 7. This compares favorably with the number of syntactic analyses, which was on average 31.2. When adding this to our experience so far that inspecting a discriminant on average does not take more time than inspecting a full syntactic structure, our method has a potential for a real gain in efficiency. Naturally, we will continue to investigate this as our treebanking effort grows.

From each subcorpus overview page there is a link to a discriminant statistics page. Figure 5 shows the top of the discriminant statistics page for the subcorpus *jh1*. First there is an overview of the number of times the different types of discriminants have been chosen. Underneath is a list of all the discriminants that have been chosen at least once. This overview is especially important for the issue of discriminant design.

The f-structure discriminants described in section 2 above described paths from PRED values to PRED values. An f-structure discriminant may also describe a path from a PRED value to an atomic value. The most often chosen discriminants on the discriminant statistics page in figure 5 are of this latter type. Originally we designed these discriminants especially for these determiners, since they are highly ambiguous function words which do not have morphology discriminants. However, since we want to avoid language specific solutions, we globally introduced f-structure discriminants from PRED values to atomic values. The resulting abundance of discriminants can be counteracted by sorting or filtering them based on frequency of use, as apparent from the statistics in figure 5.

4 Linking Disambiguation to Grammar Development

An important aspect of the TREPIL project is the link between treebanking and grammar development. In this section we briefly outline the ways in which our approach to treebanking supports grammar development.

Treebank: jh1

[size: 329, fragmented: 81, no solutions: 71, ambiguity: 377.00 (287.30), unambiguous: 8 (+47), ambiguous: 250 (-47)]

Grammar: **Norwegian bokmål** | Overview

Discriminant Types:

C(R):	C-structure rule discriminant	(56 [4 as complement])
C(C):	C-structure constituent discriminant	(2)
F:	F-structure discriminant	(232 [57 as complement])
M:	Morphology discriminant	(26 [15 as complement])

Chosen Discriminants (together 316):

Type	*Count*	*Compl*	*Discriminant*
F	21	0	'den' DET-TYPE article
F	13	0	'en' DET-TYPE article
C(R)	13	0	PP -> P YEAR
F	9	9	'ture' NTYPE NSYN common
C(R)	7	0	PROPP -> PROP N PP
C(R)	5	0	IP -> NP I'coord
C(R)	5	0	NP -> N PP
F	5	0	_TOP 'exist<[]>[]'
C(R)	4	0	FRAG -> CONJ FRAG
C(R)	4	0	QuantP -> ART NP
C(R)	4	0	ROOT -> IPimprs PERIOD

Figure 5: The Discriminant Statistics

Discriminants for LFG grammars have been developed specifically for the TREPIL project, but they were first incorporated in XLE-Web, where they complement the functionality of the native XLE interface [9]. Actual grammar writing is normally done in Emacs. But since debugging can be difficult when there are many analyses, XLE-Web provides a useful complement for grammar writers, even when treebanking is not the goal.

XLE-Web is also a valuable diagnostic resource for treebanking. When the intended analysis of a sentence is not among the analyses in the treebank, an expert annotator may wish to investigate why the intended analysis is not present. For that purpose, there is a link from the treebank sentence page which opens a browser window in XLE-Web. In XLE-Web the sentence may be modified in various ways in order to find out why it is not getting the full intended analysis. As mentioned in 3.1 above, there is a comment field for each sentence where the annotator may record such observations. All comments for a subcorpus are displayed together in the comment column on the overview page, and the text in both the sentence column and the comment column may be searched. This is useful if the annotator wants to look for similar uses of a word or phrase in other sentences, or for another instance of a similar comment.

So far we have presented treebank construction in terms of batch parsing a corpus that was given beforehand. However, the system also supports manually constructing a special editable corpus, for example a test suite. Test suites are valuable for monitoring grammar performance [6]. For that purpose, the treebank sentence page contains Add and Delete buttons which allow the annotator to man-

ually manage the composition of the corpus.

A recent innovation is our implementation of an extended version of TIGER-Search [5], which has been made accessible through a query window, cf. figure 3. Our extension consists of a generalization from tree search to DAG search, so that also f-structures can be searched. Ultimately, structural search will be a crucial feature for end users of the treebank, and we intend to improve the search interface. However, structural search can also be useful for grammar development purposes. It has been argued that only corpora that are annotated by fine-grained grammars make it possible to easily search for complex grammatical phenomena [4]. One area in which structural search should prove useful in grammar development is in diagnosing overgeneration. It is easy to incorporate a new construction found in the corpus by adding a rule to the grammar, but it is not necessarily easy to predict what effect the new rule may have on the rest of the corpus and on parsing efficiency. When the grammar writer suspects that a new rule may overgenerate, the corpus may be reparsed and reannotated. The searching facility may then be used to find all the cases in which the new rule is used. If it seems that the rule is overgenerating, the search results can be extremely helpful in finding ways to constrain the rule.

5 Conclusion

Building further on our earlier work and that of others, we are developing a treebanking toolkit that supports incremental treebanking linked to grammar development. The disambiguation is manual, but through the use of discriminants the reuse of earlier disambiguation choices in reparsing is automated.

In this paper, we have explained the motivation for sophisticated tools to support the interplay between automated processes and the human annotator, and the linking between the grammar and the treebank. We have reported on advances in our design and implementation of a treebanking toolkit. We believe that our approach is the first that has been developed for LFG grammars. Furthermore, our implementation of discriminants is independent not only of the grammar, but also of the language, and may therefore be used by any LFG grammar for any language.

References

[1] Alena Böhmová, Jan Hajič, Eva Hajičová, and Hladká Barbora. The Prague Dependency Treebank. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, chapter 7, pages 103–127. Kluwer Academic Publishers, 2003.

[2] Gosse Bouma. Treebank evidence for the analysis of PP-fronting. In *Third Workshop on Treebanks and Linguistic Theories, Seminar für Sprachwissenschaft, Tübingen, 2004*, pages 15–26, 2004.

[3] David Carter. The TreeBanker: A tool for supervised training of parsed corpora. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 598–603, Providence, Rhode Island, 1997.

[4] Dan Flickinger. Identifying complex phenomena in a corpus via a treebank lens. In *11th Annual Conference of the European Association for Machine Translation*, pages 125–129, Oslo, Norway, 2006. Oslo University.

[5] Wolfgang Lezius. Tigersearch – ein suchwerkzeug für baumbanken. In Stephan Busemann, editor, *Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002), Saarbrücken*, 2002.

[6] Stephan Oepen, Helge Dyvik, Dan Flickinger, Jan Tore Lønning, Paul Meurer, and Victoria Rosén. Holistic regression testing for high-quality MT. Some methodological and technological reflections. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation*, Budapest, Hungary, May 2005.

[7] Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christopher D. Manning. LinGO Redwoods, a rich and dynamic treebank for HPSG. In Joakim Nivre and Erhard Hinrichs, editors, *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*, pages 117–128. Växjö University Press, 2003.

[8] Victoria Rosén, Koenraad De Smedt, Helge Dyvik, and Paul Meurer. TREPIL: Developing methods and tools for multilevel treebank construction. In Montserrat Civit, Sandra Kübler, and Ma. Antònia Martí, editors, *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, pages 161–172, 2005.

[9] Victoria Rosén, Paul Meurer, and Koenraad De Smedt. Constructing a parsed corpus with a large LFG grammar. In *Proceedings of LFG'05*. CSLI Publications, 2005.

[10] Geoffrey Sampson. Thoughts on two decades of drawing trees. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, chapter 2, pages 23–41. Kluwer Academic Publishers, 2003.