TREPIL: Developing Methods and Tools for Multilevel Treebank Construction

Victoria Rosén, Koenraad de Smedt, Helge Dyvik and Paul Meurer

University of Bergen
Department of Linguistics and Comparative Literature and AKSIS
{victoria,desmedt,dyvik,paul.meurer}@uib.no

1 Introduction

Current trends in language technology require treebanks that do not stop at the level of constituent structure, but include deeper and richer levels of analysis, including appropriate meaning structures. Capturing sufficient detail at different levels of linguistic description is too complex a task to be practically achievable by manual annotation or shallow parsing; rather it requires sophisticated tools that help secure the consistency of parallel but different structures.

In conventional treebanks, grammatical functions and semantic roles are often simply attached to the syntactic constituent structure. The Penn Proposition Bank [12, 20] is basically constructed by labeling verbs as predicates and assigning appropriate semantic (thematic) roles to syntactic constituents that are in grammatical relations to the verbs. Though useful in its own right, this approach is nevertheless limited to verbs and is constrained by implicit isomorphism between the syntactic and semantic structures.

In contrast, we are constructing a multilevel treebanking tool that incorporates a deep parser and grammar for Norwegian. Inspired by the LinGO Redwoods approach [19], we are tightly linking our treebank to grammar development so as to achieve a sound embedding in grammatical theory and yield more useful results for applications.

2 The TREPIL Project

The research reported on in this paper is the first stage of the Norwegian Treebank Pilot Project (TREPIL). This project is not aimed at building a full scale treebank, but at developing a suitable methodology and sophisticated tools for the semiautomatic construction of a treebank in a later followup project.

The method is aimed at constructing a multipurpose treebank where linguistic information is represented in three distinct levels of structure:

1. constituent structure (c-structure)
2. functional structure (f-structure)
3. semantic structure (mrs-structure)

The grammatical representations are founded in Lexical-Functional Grammar (LFG) [1, 6] and the semantic representation in Minimal Recursion Semantics (MRS) [5]. Thus, our approach is not only a multilevel approach, but also integrates components from two linguistic theories. The different theories are integrated through a common grammar and lexicon.

Given the rich structural representation on three levels of linguistic description, it is not feasible to construct the treebank manually. Nor can we bootstrap from an existing treebank, as was done in the PARC 700 project [8], since there is currently no large scale treebank for Norwegian (although we should mention ongoing work at the Text Laboratory in Oslo [17, 10]). Since we have access to NorGram, a computational grammar for Norwegian [2], we will build the treebank as an automatically parsed corpus. We have constructed a treebanking toolkit consisting of NorGram in conjunction with the Xerox Linguistic Environment (XLE) [13], a large lexicon and a morphological analyzer which we have developed in cooperation with the LOGON machine translation project [18]. Furthermore, we have linked this automatic parsing system to a disambiguation module and a treebank storage system. A system for efficient treebank search still has to be developed. The treebanking toolkit is schematically represented in figure 1.

3 Multilevel Analysis with an LFG-grammar

NorGram is a large computational grammar for Norwegian Bokmål. Its core is written in the unification-based LFG formalism and it has at present 165 rules with 2,465 states, 28,990 arcs, and 125,930 disjuncts. The number of arcs corresponds to the approximate number of rules there would be in the grammar if only unary and binary-branching rules were permitted, and thus gives an impression of the approximate size of the grammar.

The preprocessing component of the grammar consists of a morphological analyzer, a compound analyzer and a named entity recognizer. The morphological analyzer is based on a lexicon containing approximately 140,000 base forms and

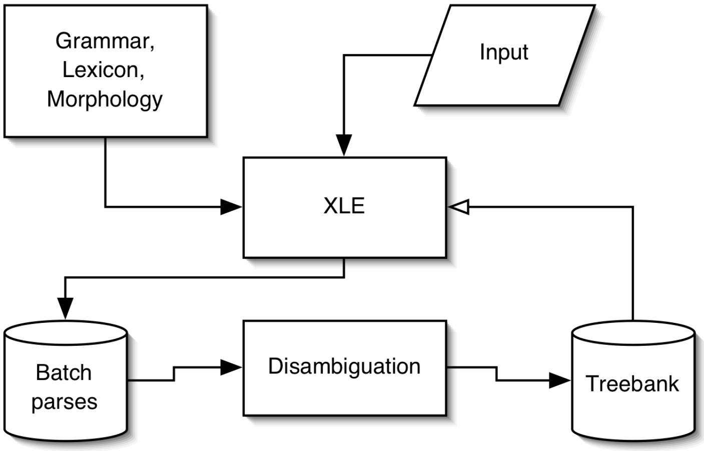


Figure 1: Diagram representing the TREPIL treebanking toolkit

1,400,000 inflectional forms. Since compounding is a highly productive process in Norwegian, not every compound can be included in the lexicon, and a means of analyzing unknown compounds on the fly is needed. Our compound analyzer uses regular expressions over strings and morphosyntactic features to derive probable segmentations for such compounds, ranking them according to number of segments and other heuristic criteria. Simplex and multiword names are recognized using a named entity recognizer which first parses the input sentence with a Constraint Grammar (CG) [11] parser for Norwegian and then applies an additional set of CG rules and a regular expression parser to extract named entities [9].

At the syntactic level, as for all LFG grammars, there are two distinct structural representations: c-structure, which is a phrase structure tree, and f-structure, which is an attribute-value matrix with information about grammatical features and syntactic functions. In figure 2 are examples of the c-structure and f-structure for sentence 1.

(1) *Petter sover ikke.*
Petter sleeps not
"Petter is not sleeping."

Unlike other LFG grammars, ours also has a semantic projection, an mrs-structure, as mentioned above. An mrs-structure is a flat structure consisting of

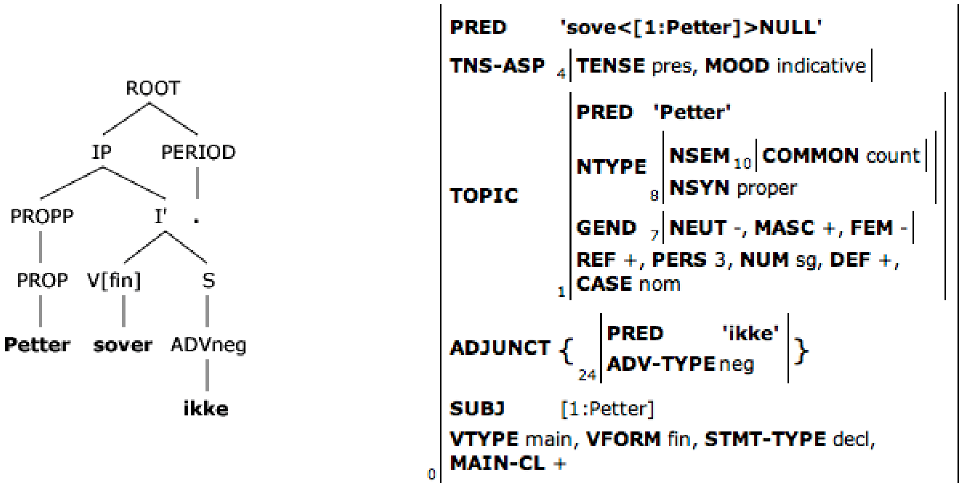
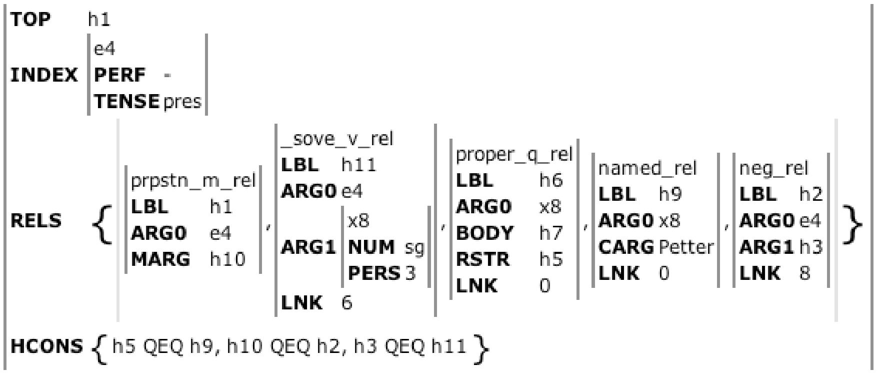


Figure 2: c-structure and f-structure for *Petter sover ikke*

a set of elementary predications (EPs), where each EP has a relation, a label or handle (LBL), and a set of argument roles (ARG0 . . . ARGn). The values of the argument roles are variables over events (*e*), individuals (*x*) or handles (*h*), and the variables may carry features expressing for example number and tense information.

Partly or completely underspecified quantifier scope is allowed by means of a QEQ relation (‘equality modulo quantifiers’) on handles. This means that there can be a single mrs-structure for a scopally ambiguous sentence. Figure 3 shows the mrs-structure for example 1, in which the quantifier *proper_q_rel* binds the variable *x8*. The restriction of the quantifier is the *named_rel* relation, which is shown by the fact that its RSTR variable is QEQ the handle of the *named_rel* EP. The body of the quantifier is not specified, which leaves open the possibility that other quantifiers in the sentence may or may not scope over it. In this example there are no other quantifiers and there is hence only one way to make the mrs-structure scopally specified: *h10 = h6, h7 = h2*.

Being derived by codescription, the mrs-structures in general may contain semantic information that cannot be derived from the c- or f-structures, which means that it is not redundant to store all three structures in the treebank. Thus, for example, while the verb is the highest predicate in the f-structure, the negation is the highest predicate in the mrs-structure (cf. that *h10 QEQ h2* and *h3 QEQ h11*).



```
TOP    h1
       e4
INDEX  PERF  -
       TENSE pres
                                     _sove_v_rel
                                     LBL   h11              proper_q_rel
         prpstn_m_rel                ARG0  e4               LBL   h6      named_rel       neg_rel
         LBL   h1                          x8               ARG0  x8      LBL  h9         LBL  h2
RELS  {  ARG0  e4                 ,  ARG1  NUM  sg       ,  BODY  h7   ,  ARG0 x8      ,  ARG0 e4   }
         MARG  h10                         PERS 3           RSTR  h5      CARG Petter     ARG1 h3
                                     LNK   6                LNK   0       LNK  0          LNK  8
HCONS { h5 QEQ h9, h10 QEQ h2, h3 QEQ h11 }
```

Figure 3: mrs-structure for *Petter sover ikke*

4 Disambiguation

For a hand-coded treebank, it is always a problem to get different annotators to annotate in the same way (e.g. Van der Beek et al. [22], Palmer et al. [20]). Even for one annotator, it can be difficult to make the same choices across different parts of the corpus. It is a great advantage of an automatically parsed corpus that the analyses will always be consistent. However, two important issues cannot be automatically resolved. One is disambiguation, the other is coverage. We have therefore paid attention to methods and tools for dealing with these problems efficiently.

A large grammar that operates on realistic sentences is bound to expose massive syntactic ambiguity. Parsing usually produces several possible analyses which may be quite numerous. State of the art approaches to disambiguation typically involve stochastic training on a treebank. If, however, there is no treebank to bootstrap from, this is not an option. A second approach is to include preferences in the grammar. In our grammar, this is done in the form of optimality marks, but these lack sufficient reliability and coverage for treebanking purposes. The solution we have chosen is manual disambiguation.

Inspection of individual candidate structures as a manual disambiguation method must be ruled out, given that the structures are quite complex and their numbers can run into the thousands. There is, however, an alternative strategy based on elementary local properties of the analyses. This technique, first proposed by Carter [4], is known as *discriminant* disambiguation. Any local property that is not shared by all analyses may be used as a discriminant. The annotator may then choose or reject properties according to whether the intended analysis should or should not have these properties. Each time the annotator makes a decision on a discriminant, the search space is diminished. Choosing a discriminant amounts to choosing all the analyses that share that property. Likewise, rejecting a discriminant amounts to rejecting all the analyses that share that property. In this way the number of remaining analyses is rapidly reduced. Discriminant disambiguation has also been used in the LinGO Redwoods treebank, and like Carter, they report that annotation by this method is very fast [4, 19].

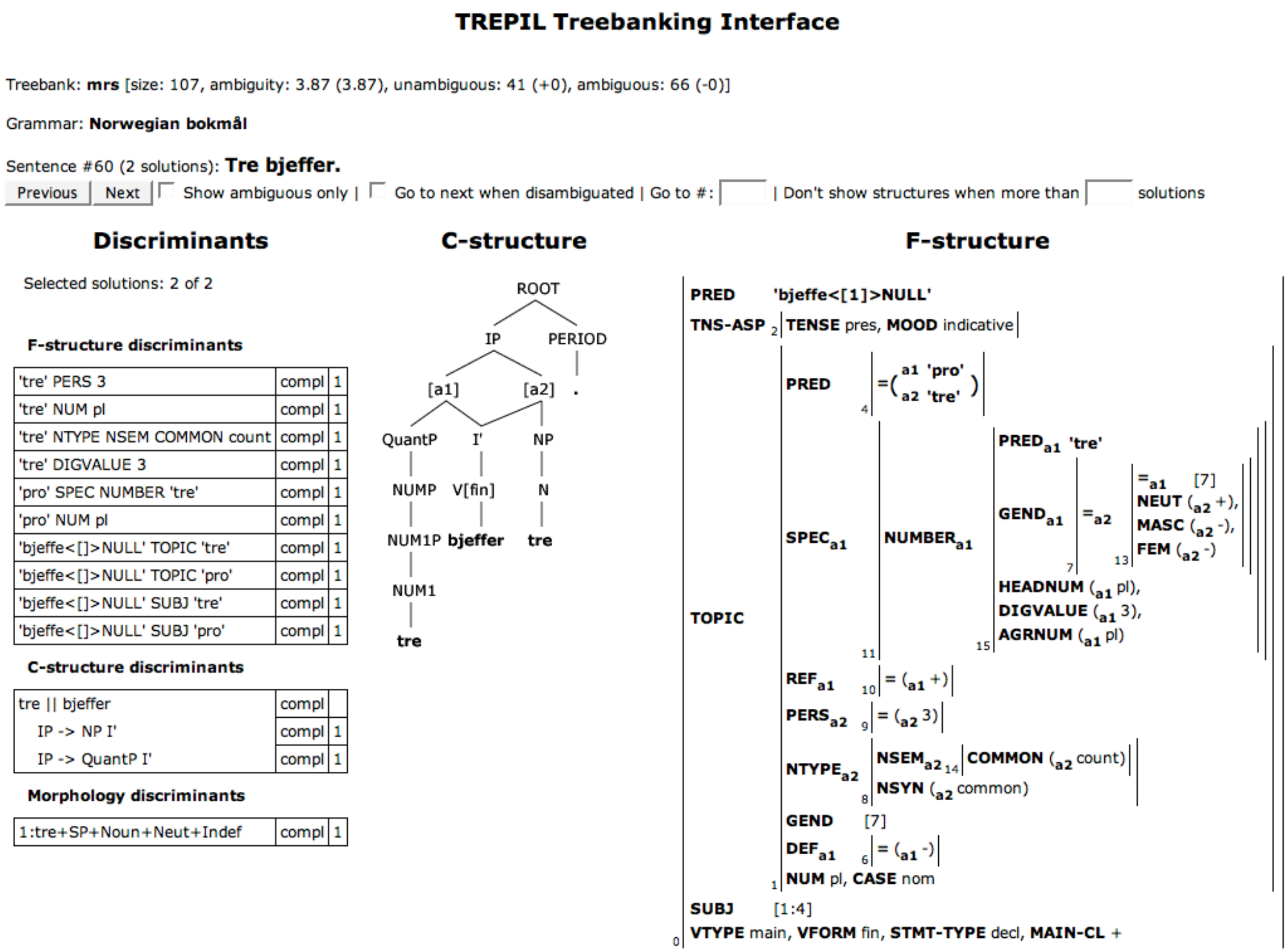
We have implemented three types of discriminants in TREPIL: c-structure discriminants, f-structure discriminants and morphology discriminants. A c-structure discriminant is the segmentation of a surface constituent string induced by a minimal subtree (a node with its immediate subnodes); in addition, the rule that gives rise to this subtree is a discriminant. An f-structure discriminant is a direct path in an f-structure from a PRED value to an embedded PRED value or from a PRED value to an atomic value. A morphology discriminant is a word with the tags it receives from morphological preprocessing. These discriminants are described in more detail in the paper "Constructing a Parsed Corpus with a Large LFG Grammar" [21]. Examples of all three types of discriminants are found in figure 4.

An interesting property of discriminant choice decisions is that they can be reused. After a revision of the grammar, each of the previously chosen discriminants for a given sentence is again applied to the revised analysis. This is possible because discriminants make no reference to the grammar rules, they apply solely to the c- and f-structures resulting from grammar application. If discriminant application again results in full disambiguation, which will be the case most of the time, no user intervention is needed. If on the other hand the discriminants have become contradictory or they no longer fully disambiguate, the annotator may revise the discriminant decisions based on a newly computed set of discriminants.

5 Treebanking Interface

Figure 4 shows a screen shot from the TREPIL Treebanking Interface. This is the first version of the annotator's tool for disambiguation. The treebanking interface is implemented in Common Lisp and uses XML, XSLT and Javascript to serve the interface web pages. C-structure trees (and graphs) are drawn using SVG (Scalable Vector Graphics). Parse and disambiguation data are stored in a relational database. The sentence to be disambiguated in this example is the one in 2.

(2) *Tre* *bjeffer.*
three/trees bark
"Three are barking." or "Trees are barking."



TREPIL Treebanking Interface

Treebank: **mrs** [size: 107, ambiguity: 3.87 (3.87), unambiguous: 41 (+0), ambiguous: 66 (-0)]

Grammar: **Norwegian bokmål**

Sentence #60 (2 solutions): **Tre bjeffer.**

Previous
Next
□ Show ambiguous only | □ Go to next when disambiguated | Go to #: | Don't show structures when more than solutions

Discriminants

Selected solutions: 2 of 2

F-structure discriminants

'tre' PERS 3	compl	1
'tre' NUM pl	compl	1
'tre' NTYPE NSEM COMMON count	compl	1
'tre' DIGVALUE 3	compl	1
'pro' SPEC NUMBER 'tre'	compl	1
'pro' NUM pl	compl	1
'bjeffe<[]>NULL' TOPIC 'tre'	compl	1
'bjeffe<[]>NULL' TOPIC 'pro'	compl	1
'bjeffe<[]>NULL' SUBJ 'tre'	compl	1
'bjeffe<[]>NULL' SUBJ 'pro'	compl	1

C-structure discriminants

tre || bjeffer	compl	
IP -> NP I'	compl	1
IP -> QuantP I'	compl	1

Morphology discriminants

1:tre+SP+Noun+Neut+Indef	compl	1

C-structure

F-structure

```
PRED     'bjeffe<[1]>NULL'
TNS-ASP 2 TENSE pres, MOOD indicative
TOPIC      PRED      =( a1 'pro'
                      4  a2 'tre' )
           SPECa1    NUMBERa1   PREDa1 'tre'
                                GENDa1   =a2   =a1  [7]
                                               NEUT (a2 +),
                                               MASC (a2 -),
                                            7 13 FEM (a2 -)
                                HEADNUM (a1 pl),
                                DIGVALUE (a1 3),
                             11 15 AGRNUM (a1 pl)
           REFa1   10 = (a1 +)
           PERSa2   9 = (a2 3)
           NTYPEa2    NSEMa2 14 COMMON (a2 count)
                    8 NSYN (a2 common)
           GEND     [7]
           DEFa1    6 = (a1 -)
         1 NUM pl, CASE nom
SUBJ     [1:4]
0 VTYPE main, VFORM fin, STMT-TYPE decl, MAIN-CL +
```

Figure 4: The TREPIL Treebanking Interface

This sentence is very simple, with only two analyses. Although the ambiguity in this sentence has its root in a lexical ambiguity, it may be disambiguated by choosing an f-structure discriminant, a c-structure discriminant or a morphology discriminant. The annotator may either choose a discriminant by clicking on it or reject a discriminant by clicking on *compl* (for ‘complement’). The number after *compl* shows how many solutions will still be left if that discriminant is accepted. For this example, the choice of any discriminant will fully disambiguate between the two analyses.

The interface also shows packed c- and f-structures. Packed structures were first implemented in XLE in order to provide a compact internal representation of the set of solutions of a sentence. The XLE display system uses this packing to simultaneously display all f-structures in one graph, and the packed f-structures in the TREPIL Treebanking Interface have been tightly modeled on XLE’s packed f-structure display. Packed c-structures, which are an innovation in TREPIL, are directed acyclic graphs, sets of c-structure trees where certain nodes that are equal across solutions are identified and where additional nodes indicate in which contexts their subnodes are valid.

As this example illustrates, there may be many more discriminants than are necessary for complete disambiguation. A topic of further research in the TREPIL project will be how the discriminants may best be displayed to the annotator in order to make disambiguation as efficient as possible. It is for instance well known that lexical ambiguities are among the easiest properties for annotators to decide on, so that it may make sense to display discriminants for lexical ambiguities first. When there are few analyses, the packed c- and f-structures may be valuable to the annotator, but when there are many analyses, these structures may be too large to even examine. Therefore the annotator may choose not to have these displayed at all when there are more than a certain number of solutions. We will experiment further with methods for optimizing the efficiency of the annotator’s task.

We could also have implemented discriminants for mrs-structures, but have chosen not to do so. There are several reasons for this decision. One is that it is not necessary; each analysis may be fully disambiguated on the basis of syntactic and lexical properties alone, since only one mrs-structure is projected from a given f-structure. Another reason is that for most annotators, c- and f-structure properties will be easier to decide on than properties of mrs-structures. Finally, and most importantly, the fact that we have based the discriminants on formal properties of c- and f-structures means that the discriminants may be used not only independently of the grammar, but also independently of the language, since all LFG grammars have c- and f-structures but only ours has an mrs-structure. Therefore, our treebanking tool may be used by any LFG grammar that is implemented in XLE.

6 Aspects of Coverage

As mentioned above, coverage is an important concern for an automatically parsed corpus. NorGram is currently being used in LOGON, a project involving machine translation from Norwegian to English in the domain of tourist texts [18]. Table 1 gives an indication of the grammar's current coverage of the LOGON development corpus.

Table 1: NorGram coverage of the LOGON development corpus (October 2005)

Item length[1]	*1–10*		*11–15*		*16–20*		*All <21*	
	#	%	#	%	#	%	#	%
Items	981		412		347		1740	
Complete parses	740	75.4	207	50.2	99	28.5	1046	60.1
Fragment parses	226	23.1	151	36.7	122	35.2	499	28.7
Total coverage	966	98.5	358	86.9	221	63.7	**1545**	**88.8**

When the parser does not find a complete parse for a sentence, it tries to produce a *fragment parse*, which represents a sentence as a structure composed of fragments which each are grammatical. There can be several reasons why a corpus sentence does not get a complete parse. In some cases fragment analysis may be intended. For instance, if the text sentence is not really a grammatical sentence, it would not be desirable to revise the grammar to allow for coverage. In other cases, fragment analysis may be preferred for reasons of parsing efficiency. For instance, allowing all types of coordination that actually occur in texts would make the parser too inefficient to be used for any practical application.

In other cases, fragment analysis may occur because the syntactic construction involved is missing from the grammar, or because a subcategorization frame for a certain verb is missing in the lexicon. In such cases, our interactive approach to treebank annotation can help us to rapidly improve coverage. We will implement a possibility for the annotator to store comments in the database so that necessary revisions to the grammar and lexicon can be implemented during the following revision cycle.

[1] Items comprise headings and other nonsentential strings in addition to sentences.

7 Conclusion

We have presented a multilevel approach to treebanking for Norwegian firmly grounded in linguistic theory through the adoption of the LFG and MRS frameworks. Nivre [15, 16] discusses the relation between treebanks and linguistic theory. He points out that it is important that the treebank representations can be converted to other representations depending on the requirements of different applications. In that sense, our treebank approach is a good starting point, since its three levels of structure contain rich grammatical and semantic information relevant for a variety of purposes.

The usefulness of semantic structures from deep parsing with LFG and MRS has been demonstrated for Norwegian in the LOGON machine translation project. Another project has used a rudimentary MRS treebanking approach with NorGram for knowledge-based anaphora resolution (Eiken [7]). A domain-specific treebank was constructed and predicate-argument relations were collected from the mrs-structures. These were subsequently used to improve preferences in anaphora resolution.

Furthermore, our grammar is not a derivative of the treebank, but will be developed in synchrony with the treebank. This contrasts with approaches aiming at distilling a grammar from a previously constructed treebank, (e.g. Cahill [3], Nakanishi et al. [14]). Even if these approaches have been successful from the narrow viewpoint of replicating constituent structure, they have so far not resulted in grammars that allow detailed projections at three distinct levels, as our NorGram grammar provides. We believe it is better to derive a treebank from a previously constructed, theoretically motivated grammar, and to further refine the grammar as needed. Our approach guarantees that the contents of the treebank are not only internally consistent, but also consistent with the grammar and all applications based on it.

References

[1] Joan Bresnan. *Lexical-Functional Syntax*. Blackwell, Malden, MA, 2001.

[2] Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. The Parallel Grammar project. In *Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation, Taipei, Taiwan*, 2002.

[3] Aoife Cahill. *Parsing with Automatically Acquired, Wide-Coverage, Robust, Probabilistic LFG Approximations*. PhD thesis, School of Computing, Dublin City University, 2004.

[4] 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.

[5] Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. Minimal Recursion Semantics: An introduction. Manuscript, in preparation.

[6] Mary Dalrymple. *Lexical-Functional Grammar*, volume 34 of *Syntax and Semantics*. Academic Press, San Diego, CA, 2001.

[7] Unni Eiken. Corpus-based semantic categorisation for anaphora resolution. Master's thesis, University of Bergen, 2005.

[8] Tracy Holloway King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. The PARC 700 dependency bank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora, held at the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03), Budapest*, 2003.

[9] Janne Bondi Johannessen, Kristin Hagen, Åsne Haaland, Anders Nøklestad, Andra Björk Jónsdottir, Dimitris Kokkinakis, Paul Meurer, Eckhard Bick, and Dorte Haltrup. Named entity recognition for the mainland Scandinavian languages. *Literary and Linguistic Computing*, 20(1):91–102, 2005.

[10] Janne Bondi Johannessen and Lars Nygaard. Oslo-skogen. En trebank for norsk. In *Rapport fra det 10. møte om norsk språk*, Kristiansand, Norway, 2004.

[11] Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttila, editors. *Constraint Grammar: A language-independent system for parsing unrestricted text*, volume 4 of *Natural Language Processing*. Mouton de Gruyter, Berlin and New York, 1995.

[12] Paul Kingsbury, Martha Palmer, and Mitch Marcus. Adding semantic annotation to the Penn Treebank. In *Proceedings of the Human Language Technology Conference, San Diego, California*, 2002.

[13] John Maxwell and Ronald M. Kaplan. The interface between phrasal and functional constraints. *Computational Linguistics*, 19:571–589, 1993.

[14] Hiroko Nakanishi, Yusuke Miyao, and Jun'ichi Tsujii. An empirical investigation of the effect of lexical rules on parsing with a treebank grammar. In Sandra Kübler, Joakim Nivre, Erhard Hinrichs, and Holger Wunsch, editors, *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories*, pages 103–126, 2004.

[15] Joakim Nivre. What kind of trees grow in Swedish soil? In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories*, pages 123–138, 2002.

[16] Joakim Nivre. Theory-supporting treebanks. In Joakim Nivre and Erhard Hinrichs, editors, *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*, pages 117–128. Växjö University Press, 2003.

[17] Lars Nygaard and Janne Bondi Johannessen. Searchtree – a user-friendly treebank search interface. In Sandra Kübler, Joakim Nivre, Erhard Hinrichs, and Holger Wunsch, editors, *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories*, pages 183–189, 2004.

[18] Stephan Oepen, Helge Dyvik, Jan Tore Lønning, Erik Velldal, Dorothee Beermann, John Carroll, Dan Flickinger, Lars Hellan, Janne Bondi Johannessen, Paul Meurer, Torbjørn Nordgård, and Victoria Rosén. Som å kappete med trollet? Towards MRS-based Norwegian–English Machine Translation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, Baltimore, MD, October 2004.

[19] Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christopher D. Manning. LinGO Redwoods, a rich and dynamic treebank for HPSG. *Research on Language & Computation*, 2(4):575–596, December 2004.

[20] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105, 2005.

[21] Victoria Rosén, Paul Meurer, and Koenraad De Smedt. Constructing a parsed corpus with a large LFG grammar. In *Proceedings of the 10th International LFG Conference (LFG'05)*. CSLI Publications, 2005.

[22] Leonoor van der Beek, Gosse Bouma, Jan Daciuk, Tanja Gaustad, Robert Malouf, Gertjan van Noord, Robbert Prins, and Begoña Villada. Algorithms for linguistic processing: NWO PIONIER progress report, August 2002. Technical report, NWO, 2002.