Parse selection with a German HPSG grammar

Berthold Crysmann[*]

Institut für Kommunikationswissenschaften, Universität Bonn &
Computerlinguistik, Universität des Saarlandes
Poppelsdorfer Allee 47
D-55113 Bonn

crysmann@ifk.uni-bonn.de

Abstract

We report on some recent parse selection experiments carried out with GG, a large-scale HPSG grammar for German. Using a manually disambiguated treebank derived from the Verbmobil corpus, we achieve over 81% exact match accuracy compared to a 21.4% random baseline, corresponding to an error reduction rate of 3.8.

1 Introduction

The literature on HPSG parsing of German has almost exclusively been concerned with issues of theoretical adequacy and parsing efficiency. In contrast to LFG parsing of German, or even to HPSG work on English or Japanese, very little effort has been spent on the question of how the intended, or, for that matter a likely parse, can be extracted from the HPSG parse forest of some German sentence. This issue becomes all the more pressing, as the grammars gain in coverage, inevitably increasing their ambiguity. In this paper, I shall present preliminary results on probabilistic parse selection for a large-scale HPSG of German, building on technology developed in the Lingo Redwoods project (Oepen et al., 2002).

The paper is organised as follows: in section 2, I shall give a brief overview of the grammar. Section 3 discusses the treebanking effort we have undertaken (3.1), followed by a presentation of the parse selection results we achieve using probabilistic models trained on different feature sets (3.2).

2 The grammar

The grammar used in the experiments reported here has originally been developed, at DFKI, in the context of the Verbmobil project (Müller and Kasper, 2000). Developed initially for the PAGE development and processing platform (Uszkoreit et al., 1994), the grammar has subsequently been ported to LKB (Copestake, 2001) and Pet (Callmeier, 2000) by Stefan Müller. Since 2002, the grammar has been extended and modified by Berthold Crysmann (Crysmann, 2003; Crysmann, 2005; Crysmann, 2007).

The grammar, codename GG, is a large scale HPSG grammar for German, freely available under an open-source license: it consists of roughly 4000 types, out of which 290 are parametrised lexical types, used in the definition of about 35,000 lexical entries. The lexicon is further extended by 44 lexical rules and about 300 inflectional rules. On the syntactic side, the grammar has about 80 phrase structure rules.

The grammar covers all major aspects of German clausal and phrasal syntax, including free word order in the clausal domain, long-distance dependencies, complex predicates, passives, and extraposition (Crysmann, 2005). Furthermore, the grammar covers different coordination constructions, including

[*] The research reported here has been carried out at the German Research Center for Artificial Intelligence (DFKI GmbH) as part of the projects COLLATE, QALL-ME, and Checkpoint, funded by the German Federal Ministery for education and Science (BMBF), the European Union, and the State of Berlin, respectively. I am also greatly indepted to my colleagues Bernd Kiefer and Günter Neumann, as well as to Stephan Oepen and Dan Flickinger for support and comments relating to the work presented here.

Proceedings of the ACL-08: HLT Workshop on Parsing German (PaGe-08), pages 9–15,
Columbus, Ohio, USA, June 2008.

the so-called SGF coordination. Furthermore, the grammar is fully reversible, i.e. it can be used for parsing, as well as generation.

The phrase structure rules of the grammar are either unary or binary branching phrase structure schemata, permitting free interspersal of modifiers between complements in the clausal domain. The relatively free order of complements is captured by means of lexical rules which permute the elements on the COMPS valence list. As a result, the verb's complements can be saturated in any order.

The treatment of verb placement is somewhat special: in sentences without a right sentence bracket, a left branching structure is assumed, permitting efficient processing. Whenever the right bracket is occupied by a non-finite verb cluster, the finite verb in the left bracket is related to the clause finla cluster by means of simulated head movement, following the proposal by (Kiss and Wesche, 1991), inter alia. As a consequence, the grammar provides both head-initial and head-final versions of the Head-Adjunct, Head-Complement and Head-Subject schemata.

As output, the grammar delivers detailed semantic representations in the form of Minimal Recursion Semantics (Copestake et al., 2005). These representations have been successfully used in the context of automated email response or question answering (Frank et al., 2006). Most recently, the grammar has been used for automatic correction of grammar and style errors, combining robust parsing with generation.

3 Parse Selection

3.1 Treebank construction

The treebank used in the experiments reported here has been derived from the German subset of the Verbmobil (Wahlster, 2000) corpus. In essence, we removed any duplicates on the string level from the corpus, in order to reduce the amount of subsequent manual annotation. Many of the duplicates thus removed were short interjection, such as *ja* "yes", *nein* "no", or *hm* "euhm", which do not give rise to any interesting structural ambiguities. As a side effect, removal of these duplicates also enhanced the quality of the resulting treebank.

The construction of the disambiguated treebank for German followed the procedure suggested for English by (Oepen et al., 2002): the corpus was first analysed with the German HPSG GG, storing the derivation trees of all successful parses. In a subsequent annotation step, we manually selected the best parse, if any, from the parse forest, using the Redwoods annotation tool cited above.

After removal of duplicates, syntactic coverage of the corpus figured at 69.3 percent, giving a total of 11894 out of 16905 sentences. The vast majority of sentences in the corpus are between 1 and 15 words in length (14757): as a result, average sentence length of parsed utterances figures at 7.64, compared to 8.72 for the entire corpus. Although average sentence length is comparatively low, the treebank still contains items up to sentence length 47.

The 11894 successfully parsed sentences have subsequently been disambiguated with the Redwoods treebanking tool, which is built on top of LKB (Copestake, 2001) and [incr tsdb()] (Oepen, 2002). Figure 2 shows the annotation of an example sentence from the treebank.

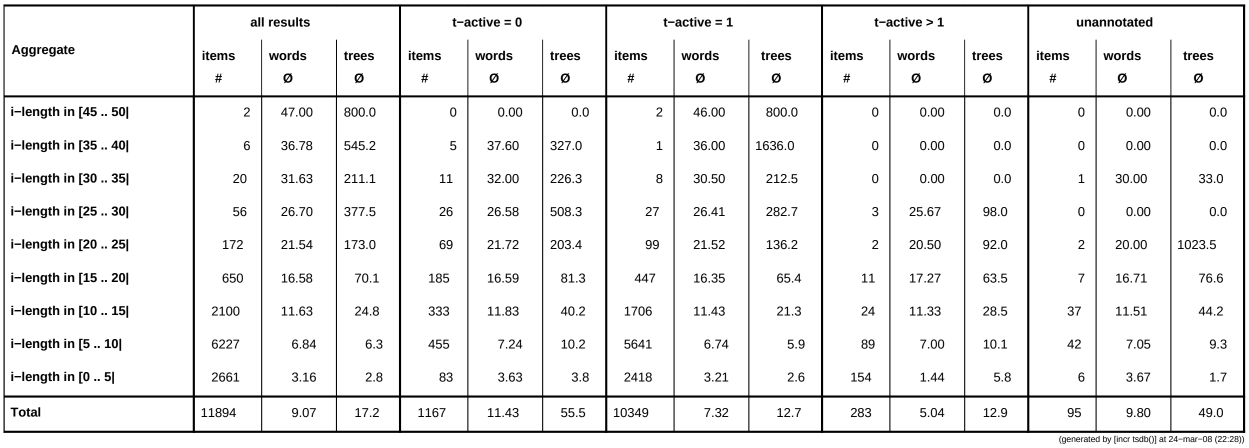
During annotation, 10356 sentences were successfully disambiguated to a single reading (87.1%). Another 276 sentences were also disambiguated, yet contain some unresolved ambiguity (2.3%), while 95 sentences were left unannotated (0.8%). The remaining 1167 items (=9.8%) were rejected, since the parse forest did not contain the desired reading. Since not all test items in the tree bank were ambiguous, we were left, after manual disambiguation, with 8230 suitable test items, i.e. test items where the number of readings assigned by the parser exceeds the number of readings judged as acceptable.

Average ambiguity of fully disambiguated sentences in the tree bank is around 12.7 trees per sentence. This corresponds to a baseline of 21.4% for random parse selection, owing to the unequal distribution of low and high ambiguity sentences.

3.2 Parse selection

3.2.1 Feature selection

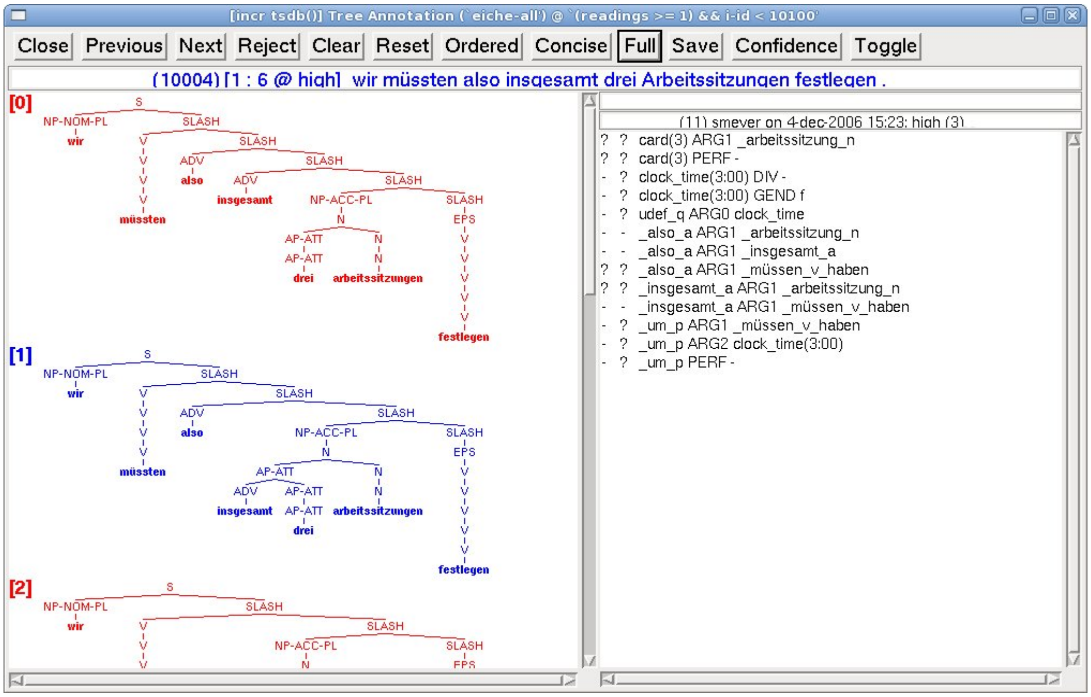
The parse selection experiments reported on here have been performed using the LOGON branch of the LKB and [incr tsdb()] systems. In particular, we used Rob Malouf's tadm maximum entropy toolkit for training and evaluation of our log-linear parse selection models.



Aggregate	all results			t–active = 0			t–active = 1			t–active > 1			unannotated		
	items	words	trees	items	words	trees	items	words	trees	items	words	trees	items	words	trees
	#	Ø	Ø	#	Ø	Ø	#	Ø	Ø	#	Ø	Ø	#	Ø	Ø
i–length in [45 .. 50|	2	47.00	800.0	0	0.00	0.0	2	46.00	800.0	0	0.00	0.0	0	0.00	0.0
i–length in [35 .. 40|	6	36.78	545.2	5	37.60	327.0	1	36.00	1636.0	0	0.00	0.0	0	0.00	0.0
i–length in [30 .. 35|	20	31.63	211.1	11	32.00	226.3	8	30.50	212.5	0	0.00	0.0	1	30.00	33.0
i–length in [25 .. 30|	56	26.70	377.5	26	26.58	508.3	27	26.41	282.7	3	25.67	98.0	0	0.00	0.0
i–length in [20 .. 25|	172	21.54	173.0	69	21.72	203.4	99	21.52	136.2	2	20.50	92.0	2	20.00	1023.5
i–length in [15 .. 20|	650	16.58	70.1	185	16.59	81.3	447	16.35	65.4	11	17.27	63.5	7	16.71	76.6
i–length in [10 .. 15|	2100	11.63	24.8	333	11.83	40.2	1706	11.43	21.3	24	11.33	28.5	37	11.51	44.2
i–length in [5 .. 10|	6227	6.84	6.3	455	7.24	10.2	5641	6.74	5.9	89	7.00	10.1	42	7.05	9.3
i–length in [0 .. 5|	2661	3.16	2.8	83	3.63	3.8	2418	3.21	2.6	154	1.44	5.8	6	3.67	1.7
Total	11894	9.07	17.2	1167	11.43	55.5	10349	7.32	12.7	283	5.04	12.9	95	9.80	49.0

(generated by [incr tsdb()] at 24–mar–08 (22:28))

Figure 1: The GG Verbmobil treebank



[incr tsdb()] Tree Annotation (`eiche-all') @ `(readings >= 1) && i-id < 10100'

(10004) [1 : 6 @ high] wir müssten also insgesamt drei Arbeitssitzungen festlegen .

(11) smever on 4-dec-2006 15:23: high (3)

```
?  ?  card(3) ARG1 _arbeitssitzung_n
?  ?  card(3) PERF -
-  ?  clock_time(3:00) DIV -
-  ?  clock_time(3:00) GEND f
-  ?  udef_q ARG0 clock_time
-  -  _also_a ARG1 _arbeitssitzung_n
-  -  _also_a ARG1 _insgesamt_a
?  ?  _also_a ARG1 _müssen_v_haben
?  ?  _insgesamt_a ARG1 _arbeitssitzung_n
-  -  _insgesamt_a ARG1 _müssen_v_haben
-  ?  _um_p ARG1 _müssen_v_haben
-  ?  _um_p ARG2 clock_time(3:00)
-  ?  _um_p PERF -
```

Figure 2: An example from the German treebank, featuring the Redwoods annotation tool

All experiments were carried out as a ten-fold cross-evaluation with 10 iterations, using 10 different sets of 7407 annotated sentences for training and 10 disjoint sets of 823 test items for testing.

The discriminative models we evaluate here were trained on different subsets of features, all of which were extracted from the rule backbone of the derivations stored in the treebank. As node labels, we used the names of the HPSG rules licensing a phrasal node, as well as the types of lexical entries (preterminals). On the basis of these derivation trees, we selected several features for training our disambiguation models: local trees of depth 1, several levels of grandparenting, i.e. inclusion of grandparent node (GP 2), great-grandparent node (GP 3) and great-great-grandparent node (GP 4), partial trees of depth 1 (+AE). Grandparenting features involve local trees of depth 1 plus a sequence of grandparent nodes, i.e. the local tree is contextualised in relation to the dominating tree. Information about a grandparent's other daughters, however, is not taken into consideration. Partial trees, by contrast, are included as a kind of back-off model.

In addition to tree-configurational features, we experimented with n-gram models, using n-gram sizes between 2 and 4. These models were further varied, according to whether or not a back-off model was included.

Apart from these linguistic features, we also varied two parameters of the maximum entropy learner, viz. variance and relative tolerance. The relative tolerance parameter restricts convergence of the model, whereas variance defines a prior in order to reduce over-fitting. In the results reported here, we used optimal setting for each individual set of linguistic parameters, although, in most cases, these optimal values figured at 10^{-4} for variance and 10^{-6} for relative tolerance.

3.2.2 Results

The results of our parse selection experiments for German are summarised in tables 1 and 2, as well as figures 3 and 4.

As our major result, we can report an exact match accuracy for parse selection of 81.72%, using great-grandparenting (GP 3) and 4-grams. This result corresponds to an error reduction by a factor of 3.8, as compared to the 21.4% random baseline.

	−AE	+AE
GP 0	77.96	78.14
GP 2	81.27	80.87
GP 3	81.34	80.4
GP 4	81.49	80.78

Table 1: PCFG model with Grandparenting

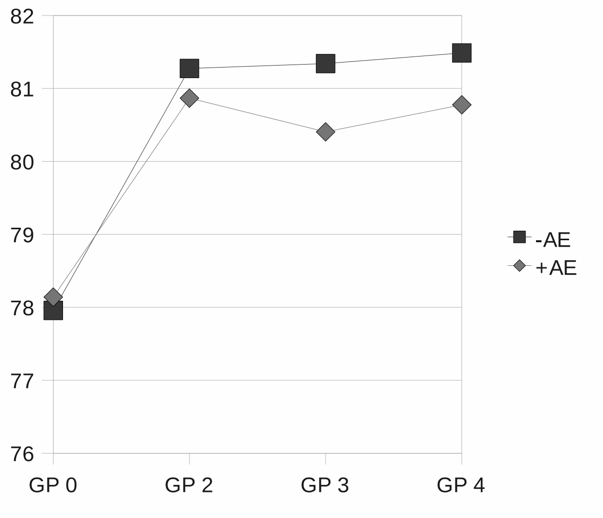


Figure 3: PCFG model with Grandparenting

Apart from the overall result in terms of achievable parse selection accuracy, a comparison of the individual results is also highly informative.

As illustrated by figure 3, models including any level of grandparenting clearly outperform the basic model without grandparenting (GP0). Furthermore, relative gains with increasing levels of grandparenting are quite low, compared to the more than 3% increase in accuracy between the GP0 and GP2 models.

Another interesting observation regarding the data in table 1 and figure 3 is that the inclusion of partial constituents into the model (+AE) only benefits the most basic model. Once the more sophisticated grandparenting models are used, partial constituent worsen rather than improve the overall performance.

Another observation we made regarding the relative usefulness of the features we have employed relates to n-gram models: again, we find that n-gram models clearly improve on the basic model without grandparenting (by about 1 percentage point), albeit to a lesser degree than grandparenting itself (see

	N0	N2	N3	N4
GP 0	77.96	78.79	78.92	78.74
GP 2	81.27	81.5	81.65	81.55
GP 3	81.34	81.44	81.51	81.72
GP 4	81.49	81.62	81.69	81.67

Table 2: PCFG model with Grandparenting & N-grams

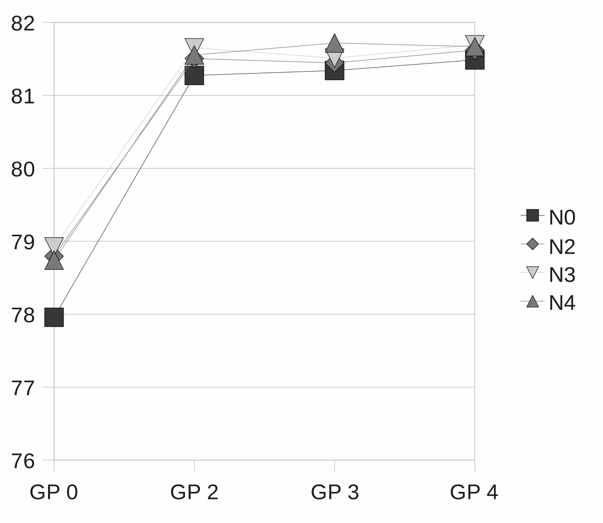


Figure 4: PCFG model with Grandparenting & N-Grams (-AE)

above). With grandparenting added, however, the relative gains of the n-gram models greatly diminishes. A possible explanation for this finding is that reference to grandparenting indirectly makes available information about the preceding and linear context, obviating the need for direct encoding in terms of n-grams. Again, the best combined model (hierarchy + n-grams) outperforms the best purely hierarchical model by a mere 0.23 percentage points. The results obtained here for German thus replicate the results established earlier for English, namely that the inclusion of n-gram information only improves overall parse selection to a less significant extent.

A probably slightly unsurprising result relates to the use of back-off models: we found that n-gram models with backing-off yielded better results throughout our test field than the correspoding n-gram models that did not use this feature. Differences, however, were not dramatic, ranging roughly between 0.07 and 0.3 percentage points.

The results obtained here for German compare quite well to the results previously achieved for the ERG, a broad coverage HPSG for English: using a similar treebank[1] (Toutanova et al., 2002) report 81.80 exact match accuracy for a log-linear model with local trees plus ancestor information, the model which is closest to the models we have evaluated here. The baseline in their experiments is 25.81. The best model they obtain includes semantic dependencies, as well, yielding 82.65 exact match accuracy.

Probably the most advanced approach to parse selection for German is (Forst, 2007): using a broad coverage LFG grammar, he reports an f-score of 83% of correctly assigned dependency triples for a reference corpus of manually annotated newspaper text. However, it is unclear how these figures relate to the exact match accuracy used here.

Relevant, in principle, to our discussion here, are also the results obtained with treebank grammars for German: (Dubey and Keller, 2003) have trained a PCFG on the Negra corpus (Skut et al., 1998), reporting labelled precision and recall between 70 and 75%. (Kübler et al., 2006) essentially confirm these results for the Negra treebank, but argue instead that probabilistic parsing for German can reach far better results (around 89%), once a different treebank is chosen, e.g. Tüba-D/Z. However, it is quite difficult to interpret the significance of these two treebank parsers for our purposes here: not only is the evaluation metric an entirely different one, but so are the parsing task and the corpus.

In an less recent paper, however, (Ruland, 2000) reports on probabilistic parsing of Verbmobil data using a probabilistic LR-parser. The parser has been trained on a set of 19,750 manually annotated sentences. Evaluation of the parser was then performed on a hold-out set of 1000 sentences. In addition to labelled precision and recall, (Ruland, 2000) also report exact match, which figures at 46.3%. Using symbolic postprocessing, exact match improves to as much as 53.8%. Table 3.2.2 summarizes Ruland's results, permitting a comparison between exact match and PARSEVAL measures. Although the test sets are certainly not fully comparable,[2] these

[1] In fact, the Redwoods treebank used by (Toutanova et al., 2002) was also derived from Verbmobil data. The size of the treebank, however, is somewhat smaller, containing a total of 5312 sentences.

[2] The overall size of the treebank suggests that we are ac-

	German
Not parsed	4.3%
Exact match	53.8%
LP	90.8%
LR (all)	84.9%
LR (in coverage)	91.6%

Table 3: Performance of Ruland's probabilistic parser (with postprocessing) on Verbmobil data

figures at least gives us an indication about how to judge the the performance of the HPSG parse selection models presented here: multiplying our 69.3% coverage with 81.72% exact match accuracy still gives us an overall exact match accuracy of 56.6% for the entire corpus.

However, comparing our German treebank to a structurally similar English treebank, we have shown that highly comparable parse selection figures can be obtained for the two languages with essentially the same type of probabilistic model.

4 Conclusion

We have presented a treebanking effort for a large-scale German HPSG grammar, built with the Redwoods treebank technology (Oepen et al., 2002), and discussed some preliminary parse selection results that are comparable in performance to the results previously achieved for the English Resource Grammar (lingoredwoods:2002tlt). Using a treebank of 8230 disambiguated sentences, we trained discriminative log-linear models that achieved a maximal exact match accuracy of 81.69%, against a random baseline of 21.4%. We further investigated the impact of different levels of grandparenting and n-grams, and found that inclusion of the grandparent node into the model improved the quality significantly, reference, however, to any higher nodes only lead to very mild improvements. For n-grams we could only observe significant gains for models without any grandparenting. We therefore hope to test these findings against treebanks with a higher syntactic complexity, in the near future, in order to establish whether these observations are indeed robust.

tually dealing with the same set of primary data. However, in our HPSG treebank string-identical test items had been removed prior to annotation and training. As a result, our treebank contains less redundancy than the original Verbmobil test suites.

References

Ulrich Callmeier. 2000. PET — a platform for experimentation with efficient HPSG processing techniques. *Journal of Natural Language Engineering*, 6(1):99–108.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan Sag. 2005. Minimal recursion semantics: an introduction. *Research on Language and Computation*, 3(4):281–332.

Ann Copestake. 2001. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford.

Berthold Crysmann. 2003. On the efficient implementation of German verb placement in HPSG. In *Proceedings of RANLP 2003*, pages 112–116, Borovets, Bulgaria.

Berthold Crysmann. 2005. Relative clause extraposition in German: An efficient and portable implementation. *Research on Language and Computation*, 3(1):61–82.

Berthold Crysmann. 2007. Local ambiguity packing and discontinuity in german. In T. Baldwin, M. Dras, J. Hockenmaier, T. H. King, and G. van Noord, editors, *Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing*, pages 144–151, Prague, Czech Republic, June. Association for Computational Linguistics.

Amit Dubey and Frank Keller. 2003. Probabilistic parsing for german using sister-head dependencies. In *ACL*, pages 96–103.

Martin Forst. 2007. Filling statistics with linguistics – property design for the disambiguation of german lfg parses. In *ACL 2007 Workshop on Deep Linguistic Processing*, pages 17–24, Prague, Czech Republic, June. Association for Computational Linguistics.

Anette Frank, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, Brigitte Jörg, and Ulrich Schäfer. 2006. Querying structured knowledge sources. *Journal of Applied Logic*.

Tibor Kiss and Birgit Wesche. 1991. Verb order and head movement. In Otthein Herzog and Claus-Rolf Rollinger, editors, *Text Understanding in LILOG*, number 546 in Lecture Notes in Artificial Intelligence, pages 216–240. Springer-Verlag, Berlin.

Sandra Kübler, Erhard W. Hinrichs, and Wolfgang Maier. 2006. Is it really that difficult to parse german? In *Proceedings of EMNLP 2006, Sydney, Australia*.

Stefan Müller and Walter Kasper. 2000. HPSG analysis of German. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 238–253. Springer, Berlin.

Stephan Oepen, E. Callahan, Daniel Flickinger, Christopher Manning, and Kristina Toutanova. 2002. LinGO Redwoods: A rich and dynamic treebank for HPSG. In *Beyond PARSEVAL. Workshop at the Third International Conference on Language Resources and Evaluation, LREC 2002*, Las Palmas, Spain.

Stephan Oepen. 2002. *Competence and Performance Profiling for Constraint-based Grammars: A New Methodology, Toolkit, and Applications*. Ph.D. thesis, Saarland University.

Tobias Ruland. 2000. Probabilistic LR-parsing with symbolic postprocessing. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 147–162. Springer, Berlin.

Wojciech Skut, Thorsten Brants, and Hans Uszkoreit. 1998. A linguistically interpreted corpus of German newspaper text. In *Proceedings of the ESSLLI Workshop on Recent Advances in Corpus Annotation*, Saarbrücken, Germany.

Kristina Toutanova, Christopher D. Manning, Stuart M. Shieber, Dan Flickinger, and Stephan Oepen. 2002. Parse disambiguation for a rich HPSG grammar. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT2002)*, pages 253–263, Sozopol, Bulgaria.

Hans Uszkoreit, Rolf Backofen, Stephan Busemann, Abdel Kader Diagne, Elizabeth Hinkelman, Walter Kasper, Bernd Kiefer, Hans-Ulrich Krieger, Klaus Netter, Günter Neumann, Stephan Oepen, and Stephen P. Spackman. 1994. Disco - an hpsg-based nlp system and its application for appointment scheduling. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING'94), August 5-9*, volume 1, pages 436–440, Kyoto, Japan.

Wolfgang Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin.