Filling Statistics with Linguistics – Property Design for the Disambiguation of German LFG Parses

Martin Forst
Institute of Natural Language Processing
University of Stuttgart, Germany
forst@ims.uni-stuttgart.de

Abstract

We present a log-linear model for the disambiguation of the analyses produced by a German broad-coverage LFG, focussing on the properties (or features) this model is based on. We compare this model to an initial model based only on a part of the properties provided to the final model and observe that the performance of a log-linear model for parse selection depends heavily on the types of properties that it is based on. In our case, the error reduction achieved with the log-linear model based on the extended set of properties is 51.0% and thus compares very favorably to the error reduction of 34.5% achieved with the initial model.

1 Introduction

In the development of stochastic disambiguation modules for 'deep' grammars, relatively much work has gone into the definition of suitable probability models and the corresponding learning algorithms. Property design, on the contrary, has rather been underemphasized, and the properties used in stochastic disambiguation modules are most often presented only superficially. This paper's aim is to draw more attention to property design by presenting linguistically motivated properties that are used for the disambiguation of the analyses produced by a German broad-coverage LFG and by showing that property design is of crucial importance for the quality of stochastic models for parse selection.

We present, in Section 2, the system that the disambiguation module was developed for as well as the initially used properties. In Section 3, we then present a selection of the properties that were expressly designed for the resolution of frequent ambiguities in German LFG parses. Section 4 describes experiments that we carried out with log-linear models based on the initial set of properties and on an extended one. Section 5 concludes.

2 Background

2.1 The German ParGram LFG

The grammar for which the log-linear model for parse selection described in this paper was developed is the German ParGram LFG (Dipper, 2003; Rohrer and Forst, 2006). It has been developed with and for the grammar development and processing platform XLE (Crouch et al., 2006) and consists of a symbolic LFG, which can be employed both for parsing and generation, and a two-stage disambiguation module, the log-linear model being the component that carries out the final selection among the parses that have been retained by an Optimality-Theoretically inspired prefilter (Frank et al., 2001; Forst et al., 2005).

The grammar has a coverage in terms of full parses that exceeds 80% on newspaper corpora. For sentences out of coverage, it employs the robustness techniques (fragment parsing, 'skimming') implemented in XLE and described in Riezler et al. (2002), so that 100% of our corpus sentences receive at least some sort of analysis. A dependency-based evaluation of the analyses produced by the grammar on the TiGer Dependency Bank (Forst et al., 2004) results in an F-score between 80.42% on all gram-

Proceedings of the 5th Workshop on Important Unresolved Matters, pages 17–24,
Ann Arbor, June 2005.

matical relations and morphosyntactic features (or 72.59% on grammatical relations only) and 85.50% (or 79.36%). The lower bound is based on an arbitrary selection among the parses built up by the symbolic grammar; the upper bound is determined by the best possible selection.

2.2 Log-linear models for disambiguation

Since Johnson et al. (1999), log-linear models of the following form have become standard as disambiguation devices for precision grammars:

$$P_\lambda(x|y) = \frac{e^{\sum_{j=1}^{m} \lambda_j \cdot f_j(x,y)}}{\sum_{x' \in X(y)} e^{\sum_{j=1}^{m} \lambda_j \cdot f_j(x',y)}}$$

They are used for parse selection in the English Resource Grammar (Toutanova et al., 2002), the English ParGram LFG (Riezler et al., 2002), the English Enju HPSG (Miyao and Tsujii, 2002), the HPSG-inspired Alpino parser for Dutch (Malouf and van Noord, 2004; van Noord, 2006) and the English CCG from Edinburgh (Clark and Curran, 2004).

While relatively much work has gone into the question of how to estimate the property weights $\lambda_1 \ldots \lambda_m$ efficiently and accurately on the basis of (annotated) corpus data, the question of how to define suitable and informative property functions $f_1 \ldots f_m$ has received relatively little attention. However, we are convinced that property design is *the* possibility of improving log-linear models for parse selection now that the machine learning machinery is relatively well established.

2.3 Initially used properties for disambiguation

The first set of properties with which we conducted experiments was built on the model of the property set used for the disambiguation of English ParGram LFG parses (Riezler et al., 2002; Riezler and Vasserman, 2004). These properties are defined with the help of thirteen property templates, which are parameterized for c-structure categories, f-structure attributes and/or their possible values. The templates are hardwired in XLE, which allows for a very efficient extraction of properties based on them from packed c-/f-structure representations. The downside of the templates being hardwired, however, is that, at least at first sight, the property developer is confined to what the developers of the property templates anticipated as potentially relevant for disambiguation or, more precisely, for the disambiguation of English LFG analyses.

The thirteen property templates can be subdivided into c-structure-based property templates and f-structure-based ones. The c-structure-based property templates are:

- cs_label <XP>: counts the number of XP nodes in the c-structure of an analysis.
- cs_num_children <XP>: counts the number of children of all XP nodes in a c-structure.
- cs_adjacent_label <XP> <YP>: counts the number of XP nodes that immediately dominate a YP node.
- cs_sub_label <XP> <YP>: counts the number of XP nodes that dominate a YP node (at arbitrary depth).
- cs_embedded <XP> <n>: counts the number of XP nodes that dominate n other distinct XP nodes (at arbitrary depth).
- cs_conj_nonpar <n>: counts the number of coordinated constituents that are not parallel at the nth level of embedding.
- cs_right_branch: counts the number of right children in the c-structure of an analysis.

The f-structure-based property templates are:

- fs_attrs <Attr$_1$ ... Attr$_n$>: counts the number of times that attributes *Attr*$_1$ ... *Attr*$_n$ occur in the f-structure of an analysis.
- fs_attr_val <Attr> <Val>: counts the number of times that the atomic attribute *Attr* has the value Val.
- fs_adj_attrs <Attr$_1$> <Attr$_2$>: counts the number of times that the complex attribute *Attr*$_1$ immediately embeds the attribute *Attr*$_2$.
- fs_subattr <Attr$_1$> <Attr$_2$> counts the number of times that the complex attribute *Attr*$_1$ embeds the attribute *Attr*$_2$ (at arbitrary depth).
- lex_subcat <Lemma> <SCF$_1$ ... SCF$_n$>: counts the number of times that the subcategorizing element $Lemma$ occurs with one of the subcategorization frames $SCF_1 \ldots SCF_n$.

- `verb_arg <Lemma> <GF>`: counts the number of times that the element $Lemma$ subcategorizes for the argument GF.

Automatically instantiating these templates for all c-structure categories, f-structure attributes and values used in the German ParGram LFG as well as for all lexical elements present in its lexicon results in 460,424 properties.

3 Property design for the disambiguation of German LFG parses

Despite the very large number of properties that can be directly constructed on the basis of the thirteen property templates provided by XLE, many common ambiguities in German LFG parses cannot be captured by any of these.

3.1 Properties that record the relative linear order of functions

Consider, e.g., the SUBJ-OBJ ambiguity in (1).

(1) [. . .] peilt [$_{S/O}$ das Management] [$_{O/S}$ ein
[. . .] aims the management a
“sichtbar verbessertes” Ergebnis] an.
“visibly improved” result at.

‘[. . .] the management aims at a “visibly improved” result.’ (TIGER Corpus s20834)

The c-structure is shared by the two readings of the sentence, so that c-structure-based properties cannot contribute to the selection of the correct reading; the only f-structure-based properties that differ between the two analyses are of the kinds `fs_adj_attrs SUBJ ADJUNCT` and `fs_subattr OBJ ADJUNCT`, which are only remotely, if at all, related to the observed SUBJ-OBJ ambiguity. The crucial information from the intended reading, namely that the SUBJ precedes the OBJ, is not captured directly by any of the initial properties. We therefore introduce a new property template that records the linear order of two grammatical functions and instantiate it for all relevant combinations. The new properties created this way make it possible to capture the default order of nominal arguments, which according to Lenerz (1977) and Uszkoreit (1987) (among others), is SUBJ, OBJ-TH, OBJ.

Similarly to the SUBJ-OBJ ambiguity just considered, the ADJUNCT-OBL ambiguity in (2) cannot at all be resolved on the basis of c-structure-based properties, and the f-structure-based properties whose values differ among the two readings seem only remotely related to the observed ambiguity.

(2) [$_{A/O}$ Dagegen] sprach sich
Against that/In contrast spoke himself
[. . .] Micha Guttmann [$_{O/A}$ für getrennte
[. . .] Micha Guttmann for separate
Gedenkstätten] aus.
memorials out.

‘In contrast, [. . .] Michael Guttmann argued for separate memorials.’ (s2090)

However, the literature on constituent order in German, e.g. Helbig and Buscha (2001), documents the tendency of ADJUNCT PPs to precede OBL PPs, which also holds in (2). We therefore introduced properties that record the relative linear order of ADJUNCT PPs and OBL PPs.

3.2 Properties that consider the nature of a constituent wrt. its function

Although linear order plays a major role in the functional interpretation of case-ambiguous DPs in German, it is only one among several ‘soft’ constraints involved. The nature of such a DP may actually also give hints to its grammatical function.

The tendency of SUBJs to be high on the definiteness scale and the animacy scale as well as the tendency of OBJs to be low on these scales has mainly been observed in studies on differential object/subject marking (see, e.g., Aissen (2003)). Nevertheless, these tendencies also seem to hold in languages like German, which does not exhibit differential object/subject marking. In (3), the indefinite inanimate DP is to be interpreted as the OBJ of the sentence and the definite human DP, as its SUBJ although the former precedes the latter.

(3) [$_{O/S}$ Nahezu stabile Preise] prognostizieren
Nearly stable prices forecast
[$_{S/O}$ die bayerischen Experten] [. . .]
the Bavarian experts [. . .].

‘The Bavarian experts forecast nearly stable prices [. . .].’ (s7357)

In order to allow these regularities to be learned from corpus data, we defined additional property templates like `isDef_<GF>` and `isHuman_<GF>`,[1] which are instantiated for all relevant grammatical functions.

3.3 Properties for the resolution of attachment ambiguities concerning extraposed constituents

A further common ambiguity in German concerns the functional attachment of extraposed constituents, such as relative clauses, *dass* clauses and infinitival VPs. In (4), e.g., there is no hard constraint that would allow us to determine whether the relative clause modifies *Rolle* or *Autoversicherung*.

(4) Eine zentrale Rolle [...] kommt der Autoversicherung zu, die ein Fünftel [...] vereinnahmt.
A central role [...] comes the car insurance to, which a fifth [...] receives.

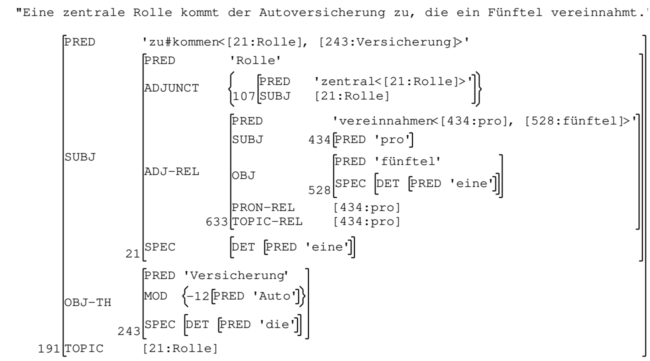
‘There is a central role for the car insurance, which receives a fifth [...].’ (s27539)

In order to allow for an improved resolution of this kind of attachment ambiguity, we introduced properties that extract the surface distance of an extraposed constituent to its functional head as well as properties that record how the functional uncertainty paths involved in these attachments were instantiated. This way, we hope to extract the information necessary to model the tendencies observed, e.g., in Uszkoreit et al. (1998).

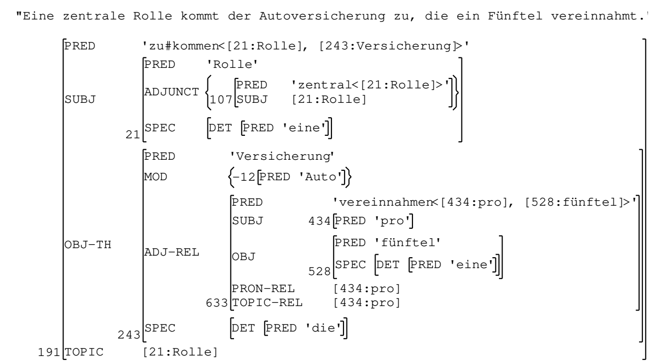
3.4 Lexicalized properties capturing dependencies

Inspired by Malouf and van Noord (2004), we finally also introduced lexicalized properties capturing dependencies. These are built on the following property templates: `DEP12_<PoS1>_<Dep>_<PoS2>_<Lemma2>`, `DEP21_<PoS1>_<Lemma1>_<Dep>_<PoS2>` and `DEP22_<PoS1>_<Lemma1>_<Dep>_<PoS2>_<Lemma2>`. These are intended to capture information on the subcategorization behavior of lexical elements and on typical collocations.

[1] Humanness information is imported from *GermaNet*.



(a) evaluated as relatively improbable due to negative weight of `DISTANCE-TO-ANTECEDENT %X`



(b) evaluated as more probable

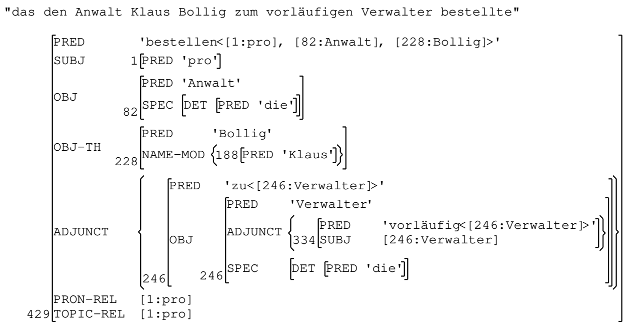
Figure 1: Competing f-structures for (4)

In the case of (5), the property `DEP21_common_Anwalt_APP_proper`, which counts the number of occurrences of the common noun *Anwalt* (‘lawyer’) that govern a proper name via the dependency `APP` (close apposition), contributes to the correct selection among the analyses illustrated in Figure 2 by capturing the fact that *Anwalt* is a prototypical head of a close apposition.[2]

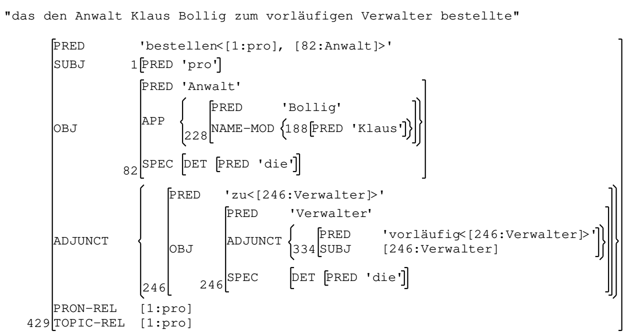
(5) [...], das den Anwalt Klaus Bollig zum vorläufigen Verwalter bestellte.
[...] which the lawyer Klaus Bollig to the interim administrator appointed.

‘[...] which appointed lawyer Klaus Bollig as interim administrator.’ (s37596)

[2] Since we have a list of title nouns available, we might also introduce a more general property that would count the number of occurrences of title nouns in general that govern a proper name via the dependency `APP`. Note, however, that the nouns that be heads of `APP`s comprise not only title nouns, but also nouns like *Abteilung* ‘department’, *Buch* ‘book’, etc.



(a) evaluated as less probable



(b) evaluated as relatively probable due to highly positive weight of `DEP21_common_Anwalt_APP_proper`

Figure 2: Competing f-structures for (5)

4 Experiments

4.1 Data

All the data we use are from the TIGER Corpus (Brants et al., 2002), a treebank of German newspaper texts comprising about 50,000 sentences. The 1,868 dependency annotations of the TiGer Dependency Bank, which have been semi-automatically derived from the corresponding treebank graphs, are used for evaluation purposes; we split these into a held-out set of 371 sentences (and corresponding dependency annotations) and a test set of 1,497 sentences. For training, we use packed, i.e. ambiguous, c/f-structure representations where a proper subset of the f-structures can be determined as compatible with the TIGER graph annotations. Currently, these are 8,881 pairs of labelled and unlabelled packed c/f-structure reprentations.

From these 8,881 pairs of c/f-structure reprentations, we extract two sets of property forests, one containing only the initially used properties, which are based on the hardwired templates, and one containing all properties, i.e. both the initially used and the newly introduced ones.

4.2 Training

For training, we use the *cometc* software by Stefan Riezler, which is part of XLE. Prior to training, however, we apply a frequency-based cutoff c to the data that ensures that a property is discriminative between the intended reading(s) and the unintended reading(s) in at least c sentences; c is set to 4 on the basis of the evaluation results achieved on our held-out set and following a policy of a 'conservative' cutoff whose only purpose is to prevent that weights be learned for sparse properties. (For a longer discussion of frequency-based cutoffs, see Forst (2007).) For the actual estimation of property weights, we then apply the combined method of incremental property selection and l_1 regularization proposed in Riezler and Vasserman (2004), adjusting the hyperparameters on our held-out set for each of the two sets of properties. In order to comparatively evaluate the importance of property selection and regularization, we also train models based on each of the two sets of properties without applying any kind of these techniques.

4.3 Evaluation

The overall results in terms of F-score and error reduction, defined as $F_\kappa = \frac{F_{actual} - F_{lower}}{F_{upper} - F_{lower}}$, that the four resulting systems achieve on our test set of 1,497 TiGer DB structures are shown in Table 1. In order to give the reader an idea of the size of the different models, we also indicate the number of properties that they are based on. All of the F-scores were calculated by means of the evaluation software by Crouch et al. (2002).

We observe that the models obtained using property selection and regularization, in addition to being much more compact than their unregularized counterparts, perform significantly better than these. More importantly though, we can see that the most important improvement, namely from an error reduction of 32.5% to one of 42.0% or from 34.8% to 51.0% respectively, is achieved by adding more informative properties to the model.

Table 2 then shows results broken down according to individual dependencies that are achieved with, on the one hand, the best-performing model based on both the XLE template-based and the newly in-

	# prop.	F-sc.	err. red.
XLE template-based properties, unregularized MLE	14,263	82.07	32.5%
XLE templ.-based pr. that survive a freq.-b. cutoff of 4, n-best grafting with l_1 regularization	3,400	82.19	34.8%
all properties, unregularized MLE	57,934	82.55	42.0%
all properties that survive a freq.-b. cutoff of 4, n-best grafting with l_1 regularization	4,340	83.01	51.0%

Table 1: Overall F-score and corresponding error reduction achieved by the four different systems on the 1,497 TiGer DB structures of our test set

troduced properties and, on the other hand, the best-performing model based on XLE template-based properties only. Furthermore, we indicate the respective upper and lower bound F-scores, determined by the best possible parse selection and by an arbitrary selection respectively.

We observe that the overall F-score is significantly better with a selection based on the model that includes the newly introduced properties than with a selection based on the model that relies on the XLE template-based properties only; overall error reduction increases from 34.5% to 51.0%. What is particularly interesting is the considerably better error reduction for the core grammatical functions `sb` (subject) and `oa` (accusative object). But also for `rcs` (relative clauses) and `mos` (modifiers or adjuncts), which are notoriously difficult for disambiguation due to PP and ADVP attachment ambiguities, we observe an improvement in F-score.

Our error reduction of 51.0% also compares favorably to the 36% error reduction on English LFG parses reported in Riezler et al. (2002). However, it is considerably lower than the error reduction of 78% reported for the Dutch Alpino parser (Malouf and van Noord, 2004), but this may be due to the fact that our lower bound is calculated on the basis of analyses that have already passed a prefilter and is thus relatively high.

5 Conclusions

Our results show that property design is of crucial importance in the development of a disambiguation module for a 'deep' parser. They also indicate that it is a good idea to carry out property design in a linguistically inspired fashion, i.e. by referring to the theoretical literature that deals with soft constraints that are active in the language for which the system is developed. Property design thus requires a profound knowledge of the language under consideration (and the theoretical literature that deals with its syntax), and since the disambiguation module operates on the output of the symbolic grammar, a good knowledge of the grammar is necessary as well.

Weighting against each other the contributions of different measures taken for improving log-linear models for parse selection, we can conclude that property design is at least as important as property selection and/or regularization, since even a completely unregularized model based on all properties performs significantly better than the best-adjusted model among the ones that are based on the template-based properties only. Moreover, property design can be carried out in a targeted way, i.e. properties can be designed in order to improve the disambiguation of grammatical relations that, so far, are disambiguated particularly poorly or that are of special interest for the task that the system's output is used for. By demonstrating that property design is the key to good log-linear models for 'deep' syntactic disambiguation, our work confirms that "specifying the features of a SUBG [stochastic unification-based grammar] is as much an empirical matter as specifying the grammar itself" (Johnson et al., 1999).

Acknowledgements

The work described in this paper has been carried out in the DLFG project, which was funded by the German Research Foundation (DFG).

Furthermore, I thank the audiences at several ParGram meetings, at the Research Workshop of the Israel Science Foundation on Large-scale Grammar Development and Grammar Engineering at the University of Haifa and at the SFB 732 Opening Colloquium in Stuttgart for their important feedback on earlier versions of this work.

References

Judith Aissen. 2003. Differential Object Marking: Iconicity vs. Economy. *Natural Language and Linguistic Theory*, 21:435–483.

gramm. relation/morphosynt. feature	upper bound F-sc.	stoch. select. all properties F-sc.	stoch. select. all properties err. red.	stoch. select. templ.-based pr. F-sc.	stoch. select. templ.-based pr. err. red.	lower bound F-sc.
all	85.50	83.01	51.0	82.17	34.5	80.42
PREDs only	79.36	75.74	46.5	74.69	31.0	72.59
`app` (close apposition)	63	60	63	61	75	55
`app_cl` (appositive clause)	53	53	100	52	86	46
`cc` (comparative complement)	28	19	-29	19	-29	21
`cj` (conjunct of coord.)	70	68	50	67	25	66
`da` (dative object)	67	63	67	62	58	55
`det` (determiner)	92	91	50	91	50	90
`gl` (genitive in spec. pos.)	89	88	75	88	75	85
`gr` (genitive attribute)	88	84	56	84	56	79
`mo` (modifier)	70	63	36	62	27	59
`mod` (non-head in compound)	94	89	29	89	29	87
`name_mod` (non-head in compl. name)	82	80	33	81	67	79
`number` (number as determiner)	83	81	33	81	33	80
`oa` (accusative object)	78	75	77	69	31	65
`obj` (arg. of prep. or conj.)	90	88	50	87	25	86
`oc_fin` (finite cl. obj.)	67	64	0	64	0	64
`oc_inf` (infinite cl. obj.)	83	82	0	82	0	82
`op` (prepositional obj.)	57	54	40	54	40	52
`op_dir` (directional argument)	30	23	13	23	13	22
`op_loc` (local argument)	59	49	29	49	29	45
`pd` (predicative argument)	62	60	50	59	25	58
`pred_restr`	92	87	62	84	38	79
`quant` (quantifying determiner)	70	68	33	68	33	67
`rc` (relative clause)	74	62	20	59	0	59
`sb` (subject)	76	73	63	71	38	68
`sbp` (logical subj. in pass. constr.)	68	63	62	61	46	55
`case`	87	85	75	83	50	79
`comp_form` (complementizer form)	74	72	0	74	100	72
`coord_form` (coordinating conj.)	86	86	100	86	100	85
`degree`	89	88	50	87	0	87
`det_type` (determiner type)	95	95	–	95	–	95
`fut` (future)	86	86	–	86	–	86
`gend` (gender)	92	90	60	89	40	87
`mood`	90	90	–	90	–	90
`num` (number)	91	89	50	89	50	87
`pass_asp` (passive aspect)	80	80	100	79	0	79
`perf` (perfect)	86	85	0	86	100	85
`pers` (person)	85	84	83	82	50	79
`pron_form` (pronoun form)	73	73	–	73	–	73
`pron_type` (pronoun type)	71	70	0	71	100	70
`tense`	92	91	0	91	0	91

Table 2: F-scores (in %) in the 1,497 TiGer DB examples of our test set

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, Bulgaria.

Stephen Clark and James R. Curran. 2004. Parsing the WSJ using CCJ and Log-Linear Models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04)*, Barcelona, Spain.

Richard Crouch, Ronald M. Kaplan, Tracy H. King, and Stefan Riezler. 2002. A comparison of evaluation metrics for a broad-coverage parser. In *Proceedings of the LREC Workshop 'Beyond PARSEVAL—Towards improved evaluation mesures for parsing systems'*, pages 67–74, Las Palmas, Spain.

Dick Crouch, Mary Dalrymple, Ron Kaplan, Tracy King, John Maxwell, and Paula Newman. 2006. XLE documentation. Technical report, Palo Alto Research Center, Palo Alto, CA.

Stefanie Dipper. 2003. *Implementing and Documenting Large-scale Grammars – German LFG*. Ph.D. thesis, IMS, University of Stuttgart. Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), Volume 9, Number 1.

Martin Forst, Núria Bertomeu, Berthold Crysmann, Frederik Fouvry, Silvia Hansen-Schirra, and Valia Kordoni. 2004. Towards a dependency-based gold standard for German parsers – The TiGer Dependency Bank. In *Proceedings of the COLING Workshop on Linguistically Interpreted Corpora (LINC '04)*, Geneva, Switzerland.

Martin Forst, Jonas Kuhn, and Christian Rohrer. 2005. Corpus-based learning of OT constraint rankings for large-scale LFG grammars. In *Proceedings of the 10th International LFG Conference (LFG'05)*, Bergen, Norway. CSLI Publications.

Martin Forst. 2007. *Disambiguation for a Linguistically Precise German Parser*. Ph.D. thesis, University of Stuttgart.

Anette Frank, Tracy Holloway King, Jonas Kuhn, and John T. Maxwell. 2001. Optimality Theory Style Constraint Ranking in Large-Scale LFG Grammars. In Peter Sells, editor, *Formal and Empirical Issues in Optimality Theoretic Syntax*, pages 367–397. CSLI Publications, Stanford, CA.

Gerhard Helbig and Joachim Buscha. 2001. *Deutsche Grammatik – Ein Handbuch für den Ausländerunterricht*. Langenscheidt, Berlin and Munich, Germany.

Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics 1999*, College Park, MD.

Jürgen Lenerz. 1977. *Zur Abfolge nominaler Satzglieder im Deutschen*. Number 5 in Studien zur deutschen Grammatik. Narr, Tübingen, Germany.

Robert Malouf and Gertjan van Noord. 2004. Wide Coverage Parsing with Stochastic Attribute Value Grammars. In *Proceedings of the IJCNLP-04 Workshop "Beyond Shallow Analyses - Formalisms and statistical modeling for deep analyses"*.

Yusuke Miyao and Jun'ichi Tsujii. 2002. Maximum entropy estimation for feature forests. In *Proceedings of the Human Language Technology Conference*, San Diego, CA.

Stefan Riezler and Alexander Vasserman. 2004. Gradient feature testing and l_1 regularization for maximum entropy parsing. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*, Barcelona, Spain.

Stefan Riezler, Tracy Holloway King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, and Mark 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, PA.

Christian Rohrer and Martin Forst. 2006. Improving coverage and parsing quality of a large-scale LFG for German. In *Proceedings of the Language Resources and Evaluation Conference (LREC-2006)*, Genoa, Italy.

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

Hans Uszkoreit, Thorsten Brants, Brigitte Krenn, Lars Konieczny, Stephan Oepen, and Wojciech Skut. 1998. Relative Clause Extraposition in German – Evidence from Corpus Studies and Acceptability Ratings. In *Proceedings of AMLaP-98*, Freiburg, Germany.

Hans Uszkoreit. 1987. *Word Order and Constituent Structure in German*. CSLI Publications, Stanford, CA.

Gertjan van Noord. 2006. At Last Parsing Is Now Operational. In Piet Mertens, Cedrick Fairon, Anne Dister, and Patrick Watrin, editors, *TALN06. Verbum Ex Machina. Actes de la 13e conférence sur le traitement automatique des langues naturelles*, pages 20–42, Leuven, Belgium.