Annotation Strategies for Probabilistic Parsing in German

Michael Schiehlen*
Institute for Computational Linguistics, University of Stuttgart
Azenbergstr. 12, D-70174 Stuttgart, Germany
Michael.Schiehlen@ims.uni-stuttgart.de

Abstract

The paper presents an unlexicalized probabilistic parsing model for German trained on the Negra treebank. Evaluation is performed with respect to constituency and dependency measures. It is observed that existing models based on Parent Encoding and Markovization optimize for constituency measures at the expense of dependency performance (at least in German). Several linguistically inspired transformation and annotation schemes are proposed which do help with dependency measures. Finally, it is shown that performance compares well with published results for German.

1 Introduction

Treebank-based probabilistic parsing has been the subject of intensive research in the past decade, resulting in models that achieve both broad coverage and high parsing accuracy (e.g. (Collins, 1999; Charniak, 2000)). For all its successes, the work on probabilistic parsing has been mostly confined to English, more specifically, to the fraction of English analysed in the Penn Treebank (Marcus et al., 1993). Recently, large treebanks have become available for languages other than English. This development has fueled interest in porting the parsing technologies developed for English and the Penn treebank format to other languages and representation formats (Collins et al., 1999; Dubey and Keller, 2003; Levy and Manning, 2003). In the wake of these endeavours it has become apparent that the settings in the English parsers are not necessarily optimal for other languages. Rather, it seems better to parameterize the parsers, so that different settings can be chosen for different languages. Considerations of modularity make a system preferable where the factors relevant in probabilistic choice are disentangled from the parsing routine. Klein and Manning (2003) have proposed and shown the promise of a system which uses a standard PCFG parser, but adeptly transforms the training corpus (and reverses these transformations on the parsing output).

In this paper, we propose a probabilistic parsing system which is quite similar in its general layout to Klein and Manning (2003)'s system. We neglect lexicalization altogether, which has been argued to be detrimental for German (Schulte im Walde, 2003; Dubey and Keller, 2003), rather we go for transformations and annotations inspired by non-treebank linguistic grammars. We evaluate all proposed factors on a development set, not only with respect to measures based on constituency (PARSEVAL), but also with dependency measures. In contrast to most other work in probabilistic parsing, we optimize for dependency rather than constituency.

Section 2 explains the difficulties peculiar to parsing German. Section 3 describes the experimental environment used. Sections 4 through 11 describe tested transformations and annotations in detail. Section 12 gives performance results on the test set. Section 13 concludes.

2 Properties of German and Negra

Two syntactic properties of German complicate context-free parsing: verbal head movement and free word-order.

2.1 Verbal Head Movement

In many German clauses, the finite verb is found in a position at the beginning of the clause while the verbs it governs are at the end. The arguments and adjuncts in between may depend on either the fronted verb or the final verbs.

2.2 Free Word-Order

Argument and adjunct NPs and PPs may be permuted at will. Quite frequently, structures of

* I would like to thank Helmut Schmid for discussions, and Kristina Spranger and an anonymous reviewer for thorough proofreading.

the type V_{fin} A_1 A_2 VP are encountered, where A_2 depends on V_{fin} and A_1 on the final VP. A context-free grammar has only one way to express such crossing dependencies: to collect all participating phrases in a single rule. Hence, a context-free grammar for German has to couch every clause frame in a separate rule. Likewise, the Negra representation format (Skut et al., 1997) treats German as a non-configurational language and uses flat structure to reduce ambiguity. It thus contains a multitude of infrequent long rules: Both the rule-per-token ratio (0.062) and the average rule length (4.93) in Negra exceed those in the Penn treebank (0.030 and 4.08).

Another effect of free word order is that the syntactic position provides at best a defeasible constraint on the argument role. For NPs, the only hard constraint on role determination is morphological case. Unfortunately, the case feature is very often ambiguous and is only disambiguated by the interaction of the agreement features (case, gender, number, and the contrast between weak and strong adjectives) in up to three words (article, adjective, noun). But even then, disambiguation may be partial, and further factors like ordering preferences (e.g. subjects usually precede objects) come into play. Since grammatical roles cannot be unambiguously derived from syntactic positions in German, any treebank for German must explicitly represent grammatical roles.

3 Experimental Setup

To facilitate comparison with previous work on German treebank parsing (Dubey and Keller, 2003; Fissaha et al., 2003; Schiehlen, 2003), we also used the Negra treebank in its bracketing format. Originally Negra does not contain trees but graphs with crossing branches. A program is, however, provided with the treebank which converts the graphs into trees with indexed traces. Negra has traces for extraposition of relative, comparative, and complement clauses, and appositions; object and VP topicalization; insertions; but not for scrambling or topicalization of subjects and sentential adjuncts, which are expressed in long rules.

We split the corpus into three subsets: Like Dubey and Keller (2003), we used sentences 18,603 through 19,602 as test set, but in contrast to Dubey and Keller (2003) we used all of them, not only the 968 sentences with a length ≤ 40. The final 400 sentences with at most 40 words were used as a development set. All other sentences constituted the training set. The development set was used to choose and calibrate the different annotations and transformations; the final results were obtained from the test set.

For our experiments, we used a probabilistic context-free grammar (PCFG) (Charniak, 1993), trained and evaluated on a treebank. In this paradigm, both the grammar and the lexicon are read off the training corpus. Each rule and each lexical entry is annotated with an expansion probability which is determined via maximum-likelihood estimates from the training corpus. We did not do any smoothing in this process (but see section 4 for the treatment of unknown words). Grammar and lexicon are then given to a standard chart parser, which produces an ambiguous representation for input sentences. Using the Viterbi algorithm, the most probable parse in terms of rule and lexical probabilities can be extracted from the chart. For these last two steps, we used an efficient CKY parser, called Bitpar (Schmid, 2004).

Instead of manipulating the parser, we followed the trail of Johnson (1998) and Klein and Manning (2003), and tried to improve models by modifying the data, i.e. the information encoded in the grammar. Thus, for each variant of the model, the training trees were annotated or transformed in some way before they were shown to the PCFG. In the PCFG output, the annotations were stripped off again and the transformations were reversed[1], so that the same test set could be used in all models.

PARSEVAL evaluation measures like labelled precision and recall were determined with the `evalb` program (Sekine and Collins, 1997). We supplement the PARSEVAL measures with f-score figures (i.e. combined precision and recall) for the task of building dependency structure (Lin, 1995), labelled by the grammatical roles of Negra. Since the PCFG parser does not output grammatical roles in all model variants, grammatical roles sometimes had to be re-introduced into the parse trees for evaluation. For each context free rule, the role interpretation most frequent in the training set was chosen.

4 Treatment of Unknown Words

The baseline model only distinguishes two classes of unknown words: those beginning with

[1] Reversal can be tricky, if added rules also occur independently in the training set. We disregarded this complication and reversed where possible.

an upper case letter and those not. These two classes are assigned all categories that uppercase and non-upper-case words may have in the corpus, weighted by frequency. This model yielded a constituency f-score of 67.2% and a dependency f-score of 78.3% on the development set.

We compared the baseline model with two other models with access to lexical knowledge. In one model (Dubey and Keller, 2003), an independently trained POS tagger (Schmid, 1994) is applied to the entire corpus (training, test, and development set). The tagging decisions are imposed on the parser. Thus the parser can learn about tagging errors in the training set, and potentially repair them in the test set. This model gave a constituency f-score of 67.3% and a dependency f-score of 77.7%.

In the third model, a morphology component (Lezius et al., 2000) was used to assign (potentially ambiguous) POS tags to all words in the test set known to this component; otherwise the baseline model was applied. This model yielded the best constituency f-score 67.4% and a dependency f-score of 78.1%. We took it as the basis for all further developments.

5 Parent Encoding and Markovization

In a standard PCFG, the basic entities with which probabilities are associated are syntax rules. Arguably, this decision is not optimal.

On the one hand, rules provide too little context. Depending on the position in the tree where a category occurs, it may have different preferences for expansion. A concise way to partially specify the tree position is to annotate every nonterminal with its parent category in the tree (Johnson, 1998). For example, a relative clause, i.e. a clause introduced by a relative pronoun, is much more likely to occur in a NP than in an AdjP or a clause. Such a preference can only be learned if the system can distinguish between relative clauses in NPs and relative clauses in AdjPs or clauses. Thus, it seems worthwhile to specify parents, or in general the most important v ancestors, with a node. A more precise way to capture the context in the higher rule is to partition according to grammatical roles rather than the mother category (Dubey and Keller, 2003). We will envisage this strategy as an alternative to parent encoding (option $v = G$ in Table 1).

On the other hand, parsers trained on treebank usually suffer from the sheer amount of mostly long and low-frequency rules provided by the treebank. Low frequency makes such rules worthless for statistical purposes. It would be much better to have a small set of short binary rules. Collins (1999) proposes to use the paradigm of Markov chains to break down rules to binary subrules. Thus, the right hand side of a rule is seen as a string that can be described by the Markov chain. An arbitrary but finite amount of context can be coded in the newly introduced intermediate categories, the 'states'. Collins (1999) proposes to let the chain begin at the head of the right hand side and encode both head and mother category in every state. Furthermore the last h categories are coded in the state, where $h \geq 0$. See Figure 1 for an example of a rule transformed in a second-order Markov model.

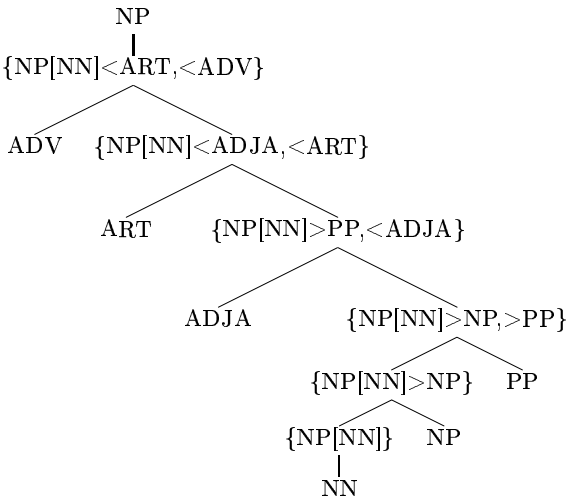

Figure 1: Markovization of the rule NP → ADV ART ADJA NN NP PP

Klein and Manning (2003) argue that Parent Encoding and Markovization should be seen as two sides of a coin: While Parent Encoding adds vertical context (from the path to the root), Markovizations cuts off horizontal context. Table 1 shows f-scores for choices of different values for v (number of ancestors stored in Parent Encoding, or grammatical roles) and h (number of siblings stored in Markovization) on the development set. A standard PCFG takes into account all siblings in a rule ($h = \infty$). The option $h \leq 2$ uses a frequency test to decide whether an intermediate category should incorporate 2 or just 1 sibling: The second-order category has to occur at least 10 times in the training cor-

Parent Encoding		Markovization				
		$h = 0$	$h = 1$	$h \leq 2$	$h = 2$	$h = \infty$
$v = 1$	no annot.	48.89	68.95	70.38	69.48	67.44
		(447)	(3639)	(5279)	(11611)	(80)
		64.9	74.4	75.0	74.6	78.1
$v = 2$	parents	51.94	71.46	72.02	70.40	67.33
		(1374)	(9355)	(11522)	(24352)	(243)
		65.6	74.9	74.6	73.7	74.9
$v = 3$	gparents	52.11	70.50	70.72	66.07	63.12
		(3827)	(20168)	(22476)	(41433)	(867)
		65.8	74.4	73.8	71.6	72.5
$v = G$	roles	49.14	68.70	68.39	66.12	62.49
		(2014)	(13132)	(15253)	(32510)	(666)
		65.5	75.2	74.8	73.2	75.1

Table 1: Markovization: Constituency F-Scores (upper), Symbol Size (middle), and Dependency F-Scores (lower) on the Development Set

pus. The constituency f-scores in Table 1 (upper lines) are by and large comparable to Klein and Manning (2003)'s results for English and the Penn Treebank: Markovization pays up to a point ($h \leq 2$), inspecting more ancestors in general is beneficial. However, performance of Parent Encoding decays rapidly with the number of ancestors, an effect that is probably due to the comparatively flat encoding in the Negra treebank.

The dependency f-scores in Table 1 (lower lines) show a radically different picture: Both Markovization and Parent Encoding actually impair performance. The reasons might be the following. The set of rules found in Negra is far from comprehensive, as only a fraction of possible word-order variations actually occurs in the training set. (Remember that each different serialization of constituents in a clause invokes a new rule.) Parent Encoding provides a fine-grained distinction of mother categories and thus aggravates rule sparseness. In contrast, Markovization aims at rule generalization, and thus would be expected to be helpful. Here, the main problem is that grammatical functions often cannot be determined locally in German, as they would have to in binary rules. We saw above that the most important feature guiding choice of grammatical role for complements, morphological agreement, is not annotated directly in Negra and not available in the PCFG. Hence, the system needs to resort to preferential regularities in positioning complements. Such preferences are, however, not accessible in a binarized (or at least an automatically binarized) version of the grammar.

All in all, we will stay with the option that performs by far best in dependency evaluation: standard PCFG ($v = 1, h = \infty$). The data shown here point at a divergence between dependency and constituency measures. Work on statistical parsing in English is typically based on the Penn treebank with rudimentary dependency information and thus geared towards optimizing the constituency measures. One might hypothesize that the availability of large-scale dependency-annotated corpora might have steered research in statistical parsing in a different direction. Furthermore, our data only show that a naive implementation of the idea of Markovization leads to performance losses; in a more sophisticated setting, Markovization might still be beneficial.

6 Treatment of Traces

Standardly, PCFG approaches suppress traces (and in general all nodes expanding to the empty string). Presence or absence of traces has no influence on the PARSEVAL measures: By definition empty nodes never lead to crossing branches. For dependency-based evaluation, traces are more important, however: Moved elements are interpreted at their base position, hence they inherit the grammatical role of their trace. We implemented the antecedent–trace relation by a slash mechanism, using the underscore symbol for slash items that are percolated upwards (cf. Figure 2).

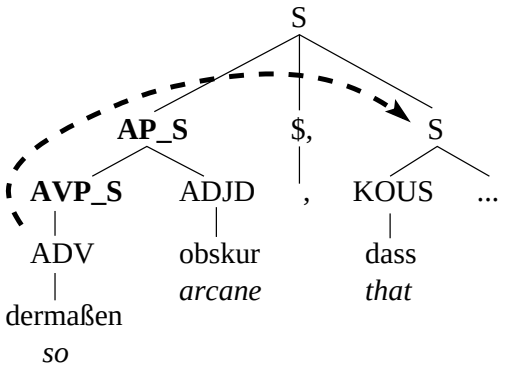

Figure 2: Slash Mechanism for Traces

This representation format led, however, to an explosion of rules and to performance losses even in dependency (see line (1) in Table 2). We followed received wisdom and removed traces.

7 Attempts at Rule Generalization

In section 5, we saw that markovization is not necessarily the best method to generalize rules.

This section describes some further attempts at smaller but linguistically equivalent grammars. Unfortunately, performance always declined with the number of rules (see Table 2, (2)–(4)), so all our attempts at generalizing rules were unsuccessful. For easy reference, annotation strategies are numbered from (1) to (19), and their performance is listed in Table 2.

(2) Coordinated Categories. Negra has special categories to distinguish coordinated phrases from base phrases. Strategy (2) conflates coordination categories and base categories.

(3) Hiding POS Tags. Many grammars make a distinction between lexical insertion rules (i.e. rules where all daughters are POS tags) and grammatical rules. The occurrence of a POS tag in a grammatical rule is only justified if the POS tag encodes rule-relevant information that is not expressible with a nonterminal category. For instance, POS tags are justified in head position as they may tell the learner something about subcategorization behaviour. Furthermore, POS tags should be used instead of nonterminals, if only a small selection of lexical classes is valid at a position. Replacing unjustified POS tags with the corresponding phrasal categories let the set of rules shrink by 22.4% on the development set.

(4) Multi-Word Lexemes. Multi-word adjectives (`MTA`), cardinal numbers (`NM`), superlatives (`AA`), proper names (`MPN`), and *zu* infinitivals (`VZ`) have special categories in Negra. Strategy (4) replaces these categories with the corresponding POS tags (`ADJA`, `CARD`, `ADJD`, `NE`, `VVIZU`); it reduced the number of rules by 2.5%.

8 Top-Down Propagation of Role Information

All in all, there is a clear correlation between categories and grammatical roles: NPs are usually arguments, PPs are arguments or adjuncts, prepositions are markers, etc. If a category occurs in a role that is not prototypical for it, it is usually somehow marked, be it lexically or syntactically. If the parser is told about the special status of the category, it can learn about such clues automatically. Thus, each annotation strategy described in this section marks a category `C` that occurs in an exceptional role `R` as `C-R`. For each strategy, we state the newly introduced category and give a short motivation.

	CF	DF	Cov	Size
baseline	67.4	78.1	100	80
(1) Traces	67.0	77.3	99.5	677
(2) Coordinated Categ.	65.2	77.9	100	73
(3) Hiding POS Tags	66.3	75.3	100	80
(4) Multi-Word Lexeme	64.4	78.1	100	78
(5) Relative Clause	69.1	79.1	100	86
(6) Adjunct NP	67.5	78.4	100	86
(7) Measure Phrase	67.6	78.1	100	82
(8) Pseudo-Genitive PP	67.6	78.6	100	84
(9) Adverbial Classif.	68.0	78.3	100	136
(10) Coordinating Item	67.7	78.1	100	89
(11) Comparative Phrase	67.7	78.2	100	89
(5)–(11)	70.0	80.1	100	172
(12) Case	71.1	81.0	100	151
(13) Verb Form	66.4	78.7	100	93
(14) Auxiliary Split	67.3	78.3	100	88
(13)–(14)	66.8	79.0	100	103
(15) Neuter Pronoun	67.0	78.0	100	81
(16) Subordinating Conj.	67.2	78.1	100	82
(17) Subcategorization	69.8	80.1	100	201
(5)–(17)	70.2	83.3	99.5	449
(18) Sentence Boundary	67.9	78.1	100	80
(19) (18)+Named Entity	69.0	79.5	100	79
(5)–(19)	71.5	84.2	99.5	445

Table 2: Evaluation on Development Set: Constituency and Dependency F-scores, Coverage, Symbol Size

(5) Relative Clauses (`S-RC`). Only relative clauses are introduced by relative pronouns.

(6) Adjunct NPs (`NP-MO`, `NN-MO`). Adjunct NPs are licensed by their head nouns (cf. also Klein and Manning (2003)'s TMP-NP strategy).

(7) Measure Phrases (`NP-AMS`). Measure phrases consist of a cardinal number and a measure noun; they occur before adjectives, adverbs (e.g. *[two months] later*), and nouns (e.g. *[two centimeters] diameter*).

(8) Pseudo-Genitive PPs (`PP-PG`, `APPR(ART)-PG`). Negra has a special role `PG` to single out PPs that alternate with postnominal genitive NPs. Such PPs are always headed by *von* (i.e. *of*).

(9) Adverbial Classification. Adverbials differ in their likelihood to modify categories like adjectives, prepositions, nouns, noun phrases, and clauses. Negra distinguishes already noun modifiers (`MNR`) from other modifiers (`MO`). We add adjective modifiers (`MAD`, e.g. *circa*), PP

modifiers (MPP, e.g. *zusammen* (i.e. *together*)), and NP modifiers (MNP, e.g. *ausgerechnet* as in *ausgerechnet Peter* (i.e. *Peter of all people*)).

(10) Coordinating Items (*-CD). Some words usually not classified as coordinating conjunctions can still act like coordinating conjunctions. Examples are the preposition *bis* (i.e. *to* in *2 to 3*), the colon in scores (*2:3*), a range of prepositions licit in constructions of the kind *arm in arm*, subordinating conjunctions that conjoin APs and other phrases (e.g. *an efficient though simple algorithm*).

(11) Comparative Phrases (*-CC). Negra analyses elliptical comparative phrases (e.g. *than Peter*) with the category of the embedded constituent (NP like *Peter*). Since the distribution of such comparative categories is clearly different, this strategy marks them with their grammatical role (CC for comparative complement).

(12) Case. As described in section 2, Negra encodes case information in grammatical roles (i.e. SB, PD for nominative, GL, GR, OG for genitive, DA for dative, OA, OA2 for accusative). As case supplies a hard constraint for the choice of NP argument roles, it seems a good idea to let the parser know about it, hence strategy (12) marks categories in case roles with the respective case. Since case is on average most constrained by the head noun, NP-internal common nouns and pronouns were also annotated with the case feature of the NP. Case was the best performing strategy (see Table 2).

9 Bottom-Up Propagation of Lexical Information

Sometimes it is helpful to subclassify terminal or nonterminal categories according to properties of their head items.

(13) Verb Form. The verb form encoded in the POS tags is passed upwards to VP and clause projections of individual verbs (cf. Klein and Manning (2003)'s SPLIT-VP).

(14) Auxiliary Split. Auxiliary verbs are subdivided into *sein* (i.e. *be*), *haben* (i.e. *have*), and *werden* (i.e. *will/be*) (cf. Klein and Manning (2003)'s SPLIT-AUX). The strategies (13) and (14) ensure that the parser can learn about the facts of verb government in German: Past participles are governed by *sein, haben, werden*. Base infinitives are governed by *werden* and modal verbs. Infinitivals with *zu* are governed by *sein, haben* and full verbs.

(15) Neuter Pronoun. The personal pronoun in neuter singular nominative or accusative (*es*, i.e. *it*) is marked as PPER.es. Only neuter singular pronouns can be used expletively.

(16) Subordinating Conjunctions. Every clause beginning with a complementizer, subordinating conjunction, or interrogative adverb is marked as S.kous.

10 Subcategorization

(17) Subcategorization. Occurrence of arguments is restricted by the subcategorization frames of verbs, nouns, and adjectives. In strategy (17), the head is marked with the arguments (and the traces of arguments) occurring in a rule, so that the parser could learn about subcategorization requirements of individual words. The frames distinguished NPs in the different cases (OG,DA,OA,OA2), predicative complements (PD), clausal complements (OC), and separated verb prefixes (SVP). Data sparseness prohibited a fine-grained classification of verb prefixes, but clausal complements were subdivided according to the verb form of their head verb (finite, infinitive, infinitive with *zu*, past participle), their complementizers (*dass* (i.e. *that*) and *ob* (i.e. *whether*)), and occurrence of a subject (infinitival clauses versus infinitival VPs).

The subcategorization information learned from the training data was supplemented with information from an independent large-scale subcategorization lexicon (Eckle-Kohler, 1999), which has frames for 16,600 verbs, 7,000 nouns, 1,800 adjectives. The subcategorization annotation scheme was among the best performing strategies, raising dependency f-score by 2% (see Table 2 and cf. similar results of Zeman (2002)).

11 Corpus Processing

(18) Sentence Boundaries. A problem intrinsic in the Negra treebank is that an automatic recognizer of sentence boundaries was applied to the data before annotation. Annotators evidently had no possibility to correct the automatic decisions. Thus, 1% of the utterances in the Negra treebank do not consist of single trees, but are collections of subsequent trees. In strategy (18), every tree was assigned its own utterance.

(19) Named Entity Recognition. A probabilistic parser is not the module of choice for named-entity recognition. We applied an independent NER module to development and test

sets, which uses information from gazetteers, but was also tuned to the training set.

12 Comparison with Previous Work

On the test set, the parsing model presented achieves a constituency f-score of 68.36% and a dependency f-score of 81.69%. Without the named entity recognition module of section 11, these figures decline to 67.37% / 81.03%. The baseline PCFG, i.e. without any annotation strategies, performs at 67.11% constituency and 77.79% dependency f-score on the test set.

The parser described compares well with other parsers, for which treebank-based evaluation results have been published. The cascaded finite-state parser of Schiehlen (2003) reaches a dependency f-score of 80.73% on entire Negra by cross validation. The lexicalized model of Dubey and Keller (2003) achieves a constituency f-score of 71.12% on nearly the same test set. All models described in this paper are unlexicalized, but the best model with regard to constituency (i.e. the $h \leq 2, v = 2$ model of section 5) outperforms Dubey and Keller (2003)'s model: On the same test set, it reaches 71.82% constituency f-score (and 76.08% dependency f-score). The results of Fissaha et al. (2003) are incomparable since they assume ideal POS tags from the treebank.

13 Conclusion

We presented a probabilistic parsing system for German trained on the Negra treebank. We tested several annotation and transformation schemes and determined a set of features that improved performance on the development set by more than 6% dependency f-score. We showed that standard techniques in probabilistic parsing like Parent Encoding (Johnson, 1998) and Markovization (Collins, 1999) are not directly applicable to German: They raise performance with respect to constituency-based measures, but not with respect to dependency-based measures. Finally we showed that our system compares well with other approaches to wide-coverage parsing in German.

References

Eugene Charniak. 1993. *Statistical Language Learning.* MIT Press.

Eugene Charniak. 2000. A Maximum-Entropy-Inspired Parser. In NAACL 1, pages 132–139.

Michael J. Collins, Jan Hajiĉ, Lance Ramshaw, and Christoph Tillmann. 1999. A statistical parser for Czech. In ACL'99, College Park, MA.

Michael J. Collins. 1999. *Head-Driven Statistical Methods for Natural Language Parsing.* Ph.D. thesis, Univ. of Pennsylvania.

Amit Dubey and Frank Keller. 2003. Probabilistic Parsing for German using Sister-Head Dependencies. In ACL'03, pages 96–103, Sapporo, Japan.

Judith Eckle-Kohler. 1999. *Linguistic knowledge for automatic lexicon acquisition from German text corpora.* Ph.D. thesis, Universität Stuttgart.

Sisay Fissaha, Daniel Olejnik, Ralf Kornberger, Karin Müller, and Detlef Prescher. 2003. Experiments in German Treebank Parsing. In *Proceedings of the 6th International Conference on Text, Speech and Dialogue (TSD-03)*, Ceske Budejovice, Czech Republic.

Mark Johnson. 1998. PCFG Models of Linguistic Tree Representations. *Computational Linguistics*, 24:613–632.

Dan Klein and Christopher Manning. 2003. Accurate Unlexicalized Parsing. In ACL'03, pages 423–430, Sapporo, Japan.

Roger Levy and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In ACL'03, pages 439–446, Sapporo, Japan.

Wolfgang Lezius, Arne Fitschen, and Stefanie Dipper. 2000. IMSLex — representing morphological and syntactical information in a relational database. In EURALEX'00, Stuttgart.

Dekang Lin. 1995. A Dependency-based Method for Evaluating Broad-Coverage Parsers. In *Proceedings of the IJCAI-95*, pages 1420–1425, Montreal.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2).

Michael Schiehlen. 2003. Combining Deep and Shallow Approaches in Parsing German. In ACL'03, pages 112–119, Sapporo, Japan.

Helmut Schmid. 1994. Probabilistic Part-Of-Speech Tagging Using Decision Trees. Technical report, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart, Germany.

Helmut Schmid. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In COLING'04, Geneva, Switzerland.

Sabine Schulte im Walde. 2003. *Experiments on the Automatic Induction of German Semantic Verb Classes.* Ph.D. thesis, IMS Stuttgart.

Satoshi Sekine and Michael Collins. 1997. EVALB - bracket scoring program. retrievable from: http://cs.nyu.edu/cs/projects/proteus/evalb/.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An Annotation Scheme for Free Word Order Languages. In ANLP'97, Washington, DC.

Daniel Zeman. 2002. Can Subcategorization Help a Statistical Dependency Parser? In COLING'02, Taipei, Taiwan.