# Parsing Without Grammar – Using Complete Trees Instead

**Sandra Kübler**
Seminar für Sprachwissenschaft
Universität Tübingen
Wilhelmstr. 19
72074 Tübingen, Germany
kuebler@sfs.uni-tuebingen.de

## Abstract

In recent years, research in parsing has concentrated on weakening the locality of parsing decisions: history-based parsing (Black *et al.* 92) uses previous parse decisions on which they condition the current decision.

We propose a new approach that combines the use of complete trees with an efficient processing based on Memory-Based Learning (Stanfill & Waltz 86; Daelemans *et al.* 99). In this approach, parsing consists of the selection of the most similar syntactic tree in the training data. This tree is then adapted to the input sentence.

The definition of similarity between sentences is formulated on the levels of words, POS tags, and chunks (Abney 91; Abney 96).

The evaluation of this approach shows that while precision and recall based on the PARSEVAL measures (Black *et al.* 91) do not reach state of the art parsers yet (F1=87.19 on syntactic constituents, F1=77.78 including function-argument structure), the parser shows a very reliable performance where function-argument structure is concerned (F1=96.52). The lower F-scores are very often due to unattached constituents.

## 1 Introduction

Many recent developments in parsing are based on the assumption that pure context-free rules, even extended by probabilities, are too local in nature to cover natural language phenomena. These new approaches (cf. e.g. (Collins 99; Charniak 01) generally apply some definition of *history-based grammar* (Black *et al.* 92) and condition parsing decisions on previous decisions. Ideally, each parsing decision would be conditioned on the parse tree of a sentence that has already been generated and on all parsing decisions made to this point in the analysis. Since the amount of training data is restricted, the models need to be restricted so that only the most informative type of information is used in determining rule probabilities. The types of information considered relevant generally include information on the heads of phrases and on their mother nodes. However, these types of information still model only a slightly less local context in a syntax tree.

*Data-Oriented Parsing* (DOP) (Bod & Scha 96; Bod 98) is the first approach that relies on larger structures that go beyond mere context-free rewrite rules instead of using more parameters. Instead, DOP uses tree fragments of differing size in the search for the best parse tree. However, since the search for the best parse tree cannot be reduced to the search for the best derivation, most DOP models are computationally intensive.

We will propose here a new parsing strategy, *Memory-Based Parsing (MBP)*, that uses complete trees as its repository of grammatical information. Instead of constructing a parse tree from different fragments, our algorithm chooses the most similar tree in an instance base and adapts this tree to the input sentence. The choice of the most similar tree is guided by POS and chunk (cf. (Abney 91; Abney 96)) information from preprocessing steps. Previous versions of this system were presented in (Kübler & Hinrichs 01a; Kübler & Hinrichs 01b), a more detailed description of the parser can be found in (Kübler 02).

The remainder of this paper is structured as follows: In section 2, we will give a short introduction to the problems of parsing a non-configurational language such as German, and in Section 3, we will give a short description of the treebank used for testing. In section 4, we will present our algorithm in more detail, section 5 provides an evaluation of this approach on German data, and in section 6, we will conclude and describe future work.

## 2 Parsing Non-Configurational Languages

The parsing algorithm presented here is based on the machine learning paradigm of *Memory-Based Learning* (MBL) (Stanfill & Waltz 86; Daelemans *et al.* 99). MBL uses an instance base of previ-

ously encountered instances, which are acquired during training as the repository of knowledge. The classification of a new instance is performed by searching for the most similar instance in the instance base. Thus, the success of such an approach crucially depends on the selection of the similarity function and the selection of the feature set on which the similarity function performs the comparison of instances.

Since MBL is a propositional learning method, previous approaches to parsing (cf. e.g. (Buchholz 02; Tjong Kim Sang 01)) were defined as a cascade of classifiers each using as input the output of the previous classifier in the cascade and classifying a very restricted subset of the intended linguistic structures. Such an approach, however, assumes a general independence of decisions within one classifier and to a great extent also between levels. Such an approach is feasible for highly configurational languages such as English, in which the ordering of constituents gives very reliable information about the grammatical function of a constituent. Thus, the assignment of grammatical functions to constituents can generally be based on a very local context. For non-configurational languages such as German, however, more non-local information is needed for this task. The decision that the constituent *das* in the sentence *das überlasse ich ganz Ihnen* (that I will completely leave up to you) is the direct object and not the subject cannot be derived from the knowledge that this constituent is the first constituent of the sentence[1]. Here, the parser needs to base the decision on the knowledge that the verb is followed by the noun phrase *ich*, which is morphologically marked as nominative.

Such problems, as well as problems resulting from the high degree of long-distance relations in German, can only be solved in a systematic way if larger tree fragments are taken into account. Therefore, *MBP* has as its task the search for the most similar (complete) tree in the instance base. By selecting complete trees, the parser finds the syntactic annotations for complete sentences including long-distance relations. But before the parser is introduced, we will give a short introduction to the corpus used in this study, TüBa-D.

[1] Morphological information does not support the decision since *das* is ambiguous between nominative and accusative.

## 3 The Tübingen Treebank for German, TüBa-D

The treebank used for evaluation is the TüBa-D treebank (Stegmann *et al.* 00; Hinrichs *et al.* 00a; Hinrichs *et al.* 00b). This treebank was developed in the context of VERBMOBIL, a long-term Machine Translation project funded by the German Ministry for Education, Science, Research, and Technology (BMBF). VERBMOBIL had as its goal the speaker-independent translation of spontaneous speech in the domains business appointments, travel scheduling, and hotel reservations. As a consequence, the TüBa-D treebank contains transliterations of spontaneous dialogues in the above mentioned domains. The transliterated data exhibit all the characteristics of spontaneous speech, including hesitation noises, false starts, interruptions, repetitions, and fragmentary utterances.

TüBa-D consists of more than 38 000 annotated sentences. In order to give a more realistic picture of the size of the treebank concerning the definition of sentence in a speech corpus, we counted the number of separate trees that were assigned to these sentences: approximately 67 000 trees. The sentences in TüBa-D tend to be rather short (the average sentence length is: 4.5 words), the challenge for the parser is based more on speech phenomena such as ungrammaticality or atypical constituent order than on extremely complex syntactic phenomena. The following sentence, which was taken from the treebank, is typical in this respect: *ich habe hier übrigens auch schon mir Unterlagen zuschicken lassen von verschiedenen Hotels* (by the way here I have also had brochures sent to me about different hotels already). Here, the modifiers *hier übrigens auch schon* precede the personal pronoun dative object *mir*, and the modifier of the accusative object, *von verschiedenen Hotels*, is extraposed.

For the annotation of the treebank, a theory-neutral and surface-oriented annotation scheme has been adopted that is based on the notion of *topological fields* (cf. e.g. (Herling 21; Höhle 86)) and enriched by a level of predicate-argument structure. The linguistic annotations pertain to the levels of morpho-syntax (POS tagging), syntactic phrase structure, and function-argument structure.

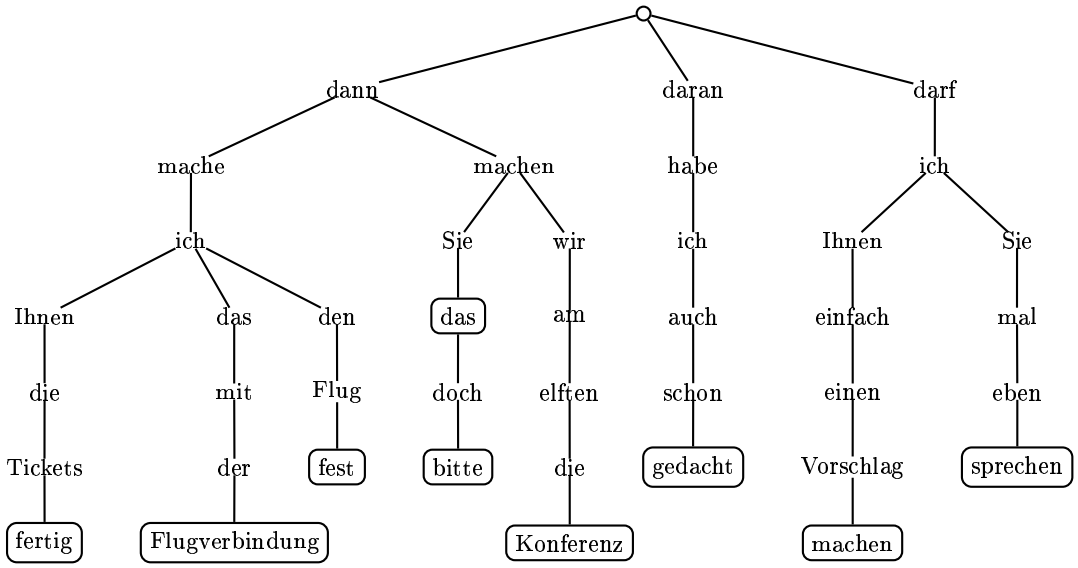


Figure 1: The prefix trie for nine sentences.

## 4 Parsing Function-Argument Structure Using an Instance Base of Trees

The parser *MBP* is divided into two main modules: In the first module, information about the sequences of words and POS tags is used to find the most similar sentence in the instance base. If this module fails to retrieve a reasonably similar sentence, the input sentence is passed to the backing-off module, which relies mainly on chunk information. The latter module is also responsible for the annotation of phrases that were omitted (cf. below) in the first module.

### 4.1 The Search Module Based on Words and POS Tags

Since the selection of the most similar tree for an input sentence needs to be based on the complete sentence rather than on a window of a fixed length, as used in cascaded parsing architectures, the search in the instance base needs to be adapted to the requirement that a flexible number of features serves as the basis for the similarity function. Thus, *MBP* uses a prefix trie of words as its instance base, and the search for the most similar tree is equivalent to the search for the most similar sequence of words in the trie. Figure 1 shows a sample of such a prefix trie for nine sentences. The final words of the sentences are marked in circles.

If a word is not found at a specific node in the trie, the search algorithm allows ignoring words or phrases either in the input sentence or in the sentence from the instance base. The phrase boundaries for the input sentence are determined in a pre-processing step via the use of a chunk parser (Abney 91; Abney 96)[2]. Thus, the input sentence *dann mache ich die Tickets fertig* (then I will arrange the tickets) will lead to the sentence *dann mache ich Ihnen die Tickets fertig* (then I will arrange the tickets for you). The additional noun phrase in the input sentence, *Ihnen*, is omitted in the search. The tree structure that is attached to this sentence is used for assigning a tree structure to the input sentence.

Now the parser is faced with the problem that the most similar sentence that it found in the instance base is not identical to the input sentence. For this reason, the retrieved structure needs to be adapted to the input sentence. There are many different possibilities how such an adaptation might be approached. However, we have made the decision to allow only the most conservative modification: the omission of the phrases or words that were omitted in the search through the trie. Thus, the structure for the sample input sentence needs to be modified so that the additional noun phrase is removed. The tree structure is shown in Figure 2, the phrase to be re-

[2] *MBP*, however, does not only extend the chunk parses with function-argument structure. It rather uses the chunk parse as additional information for the selection of the most similar sentence. There are significant differences in the annotation of phrase boundaries between a chunk parse and a complete syntactic structure due to the deterministic nature of the chunk parser.

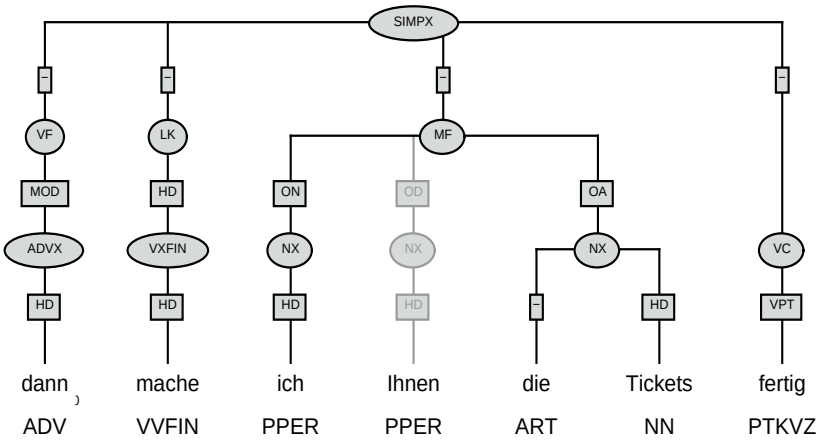


Figure 2: The most similar tree structure for the input sentence *dann mache ich die Tickets fertig*. The phrase to be removed is shown in gray.

moved shown in gray. The sentences is structured into phrases (e.g. NX for "noun phrase" and VXFIN for "finite verb phrase"), topological fields (e.g. VF for "initial field" and LK for "left sentence bracket"), and clauses (SIMPX for "simplex clause"). The edge labels denote head/non-head distinctions within clauses and the grammatical functions of phrases (e.g. ON for "subject", OD for "dative object", OA for "accusative object", and MOD for an "underspecified adjunct")[3].

Chunks which could not be attached to the sentence are then looked up separately. They receive a phrasal annotation, but no attempt is made to attach them to the sentence.

This search strategy as described above is prone to allow too many phrases or words in the input sentence or in the training sentences to be omitted. In the worst case, the most similar sentence might be selected based on the identity of a single word. Another problem consists in the fact that phrases can be omitted which are crucial for the structure of the sentence. It would be rather unfortunate if the word *Flieger* were omitted in the sentence *es gehen stündlich Flieger ab München* (there go hourly planes from Munich).

These problems are avoided to a high degree by the introduction of cost-based weights. Each word or phrase is assigned a weight (between 0 and 1) which describes the "importance" of the item for the sentence. The weights are calculated during training based on the relative frequency of the POS tag sequences with and without the item and on the degree to which the tree structure needs to be modified if the item is omitted. Thus, the word *Flieger* will have a higher weight than the phrase *ab München*. At each point where the parser has a choice of different actions, it selects the analysis which has accumulated the lowest overall weight.

### 4.2 The Backing-Off Module

The search strategy described in the previous section is very well suited for finding analyses for sentences that have closely related equivalents in the instance base. Due to the limited size of any treebank, this search strategy will fail for many sentences. In such cases, the parser backs off to a more robust module, which applies less conservative search strategies. The most important strategy in this module is the search for chunk sequences. For this reason, the input sentence is sent to the chunk parser[4] in a preprocessing step. A chunk parser is a cascade of deterministic finite-state transducers. This allows a very robust and efficient first analysis of the phrasal and clausal structure of a sentence. The input sentence *ab Donnerstag bin ich wieder hier* (from Thursday on I will be here again) receives the chunk structure shown in Figure 3.

The sentences in the instance base receive the same annotation during the training phase. Thus, the search for identical chunk sequences will be successful for sentences such as *ab Donnerstag dem dritten bin ich wieder hier* (from Thursday the third on I will be here again) or *nach einer langen Woche sind Sie dann zurück* (after a long week you will be back then). Since the internal

[3]For a more thorough introduction to the annotation scheme cf. (Stegmann *et al.* 00).

[4]We use Steve Abney's implementation CASS (Abney 91; Abney 96).

```
[simpx
    [px
       [appr ab]
       [nx2
          [day Donnerstag]]]
    [fcop bin]
    [nx4
       [pper ich]]
    [advx
       [adv wieder]]
    [advx
       [adv hier]]]
```

Figure 3: The chunk structure for the sentence *ab Donnerstag bin ich wieder hier.*

structure of the chunks in the input and the most similar sentence may be different, the structure of the most similar sentence needs to be adapted to the input sentence. In order to achieve this, the phrases on the tree structure which correspond to a specific chunk need to be identified and, if they differ in structure from the input sentence, replaced by the correct phrases. While finding correct phrase annotations in the instance base is quite straightforward, the identification of phrases corresponding to chunks is not: a chunk may correspond to more than one phrase. In such cases, several phrases in the tree structure may have to be replaced.

## 5 Evaluation

*MBP* is a very efficient parser since it is mostly deterministic. The parser forgoes a complete search of the search space. It rather applies a best-first search strategy. In order to minimize the loss in quality, it uses a wider context than most other parsing algorithms.

The time and memory requirements were tested on an Athlon 1700+ (256 MB memory). *MBP* used approximately 82 MB of memory in training and 52 MB in testing. Training with all 67 000 sentences lasted 447 seconds, a test with 6 697 sentences took 115 seconds, including POS tagging and chunking. In other words, *MBP* parsed more than 58 sentences per second.

We evaluated *MBP* on the approximately 67 000 trees of TüBa-D (cf. Section 3). In order to minimize the effects of benign data splitting, we used *ten-fold cross validation* (Weiss & Kulikowski 91): the set of 67 000 trees is randomly split into ten folds. From these annotated trees, the sequences of words were extracted. With these data, ten experiments were performed, each using one of the folds for testing and the remaining nine folds for training. The results given below were averaged over the results of the ten test runs.

For every test run, the sentences extracted from the treebank annotations were first sent through preprocessing, i.e. POS tagging using TnT (Brants 00) and chunk parsing using CASS. Using the information from preprocessing, *MBP*, assigned complete analyses to the sentences. From the results, we calculated the PARSEVAL measures *labeled precision* and *labeled recall*, both based solely on syntactic categories, as well as *(function-) labeled recall* and *(function-) labeled precision*, based on syntactic categories and grammatical functions.

The results of the evaluation are shown in Table 1. The first three metrics concern recall, precision, and the F-score based on the syntactic categories of the constituents (ignoring the functional labeling) while the next three figures give the performance of the parser when a combination of syntactic and functional labels are evaluated. These measures, however, do not make a distinction whether a constituent receives the wrong grammatical function or whether the constituent could not be attached by the parser. In both cases, the combination of syntactic and functional labels for the gold standard and the parser's output are not identical and are thus counted as incorrect. An unattached constituent, however, leads to at least one more incorrect constituent since the node that should dominate this constituent consequently shows the wrong yield. For this reason, we also give the percentage of unattached constituents, i.e. the percentage of root nodes in the parser output which have an equivalent node in the gold standard, but which is not a root node. These numbers show that a considerable number of constituents could not be attached.

For both types of evaluation, the syntactic evaluation as well as the evaluation on both syntactic and functional labels, the numbers for precision are considerably higher than for recall. This, in combination with the high percentage of unattached constituents, is an indication that the parser could not find a spanning analysis for sentences in all cases. If, however, such a spanning

| | |
|---|---|
| labeled recall (syntactic categories) | 82.45% |
| labeled precision (syntactic categories) | 87.25% |
| $F_1$ | 84.78 |
| labeled recall (incl. functional categories) | 71.72% |
| labeled precision (incl. functional categories) | 75.79% |
| $F_1$ | 73.70 |
| unattached constituents in recall | 7.14% |
| unattached constituents in precision | 7.60% |
| functional recall of attached constituents | 95.31% |
| functional precision of attached constituents | 95.21% |
| $F_1$ | 95.26 |

Table 1: *MBP*'s results for the TüBa-D treebank.

analysis was found, the parse is reliable.

Additionally, the last two lines in Table 1 give an evaluation of the functional labeling of constituents that could be attached to a higher constituent. We have previously explained that unattached phrases result in low precision numbers since the unattached root node does not contain the correct grammatical function. Figure 4 gives an example for a partial analysis for the sentence *also mindestens zwei Treffen denke ich müßten wir noch abhalten* (so I think we still need to have at least two more meetings).

Here, the parser could not attach the noun phrase *mindestens zwei Treffen* to the remainder of the sentence *müßten wir noch abhalten*. When analyzing precision based on syntactic categories and grammatical functions, the root node of the noun phrase would count as incorrect since it does not carry the grammatical function OA, which it would be assigned if it were attached. In the analysis shown in the last two lines in Table 1, we have consequently only counted nodes that are dominated by another node, i.e. non-root nodes. Therefore, the root node of the noun phrase is not considered in the calculation, and for this example, the only incorrect grammatical function is V-MOD, which should correctly be analyzed as MOD.

In order to test the generalization capacity and correctness of the two modules of the parser, we tested each module separately. In the case of the trie search, this means that if the module did not find any similar enough sentence in the instance base, the input sentence was split up into chunks, and the chunks were looked up and annotated, but there was no attempt to group these separate phrases into a tree. In the case of the backing-off module, the sentences were sent directly to this module.

The results of these test runs are shown in Table 2. It is evident that the figures for the trie-based search are significantly lower that for the complete parser, which shows that this module has a very restricted coverage. The results for the backing-off module, on the other hand, are slightly higher than for MBP. This is counterbalanced, however, by the lower quality of functional labeling in the cases where the trie-based module was successful in finding a similar sentence in the instance base (cf. rows 4-8 in Table 2). Thus, the combination of both modules gives the best balance between coverage and quality.

In order to determine the influence of the rather limited amount of training data, we also performed a leave-one-out test for 5 000 sentences. For this test, one sentence was randomly removed from the treebank, the parser was trained on the remaining treebank and tested on the single sentence. Since training takes approximately 6 minutes, this test had to be restricted to 5 000 cycles. The results in Table 3 are quite promising. It is obvious that both precision and recall increased significantly, which indicates that the performance of the parser would increase even further if more syntactically annotated training data were available.

## 6 Conclusion and Future Work

*MBP* is a memory-based parser that is based on a new approach to parsing: instead of building the intended tree structure incrementally such as stochastic parsers or other memory-based approaches to parsing do, *MBP* attempts to find the most similar sentence in the instance base

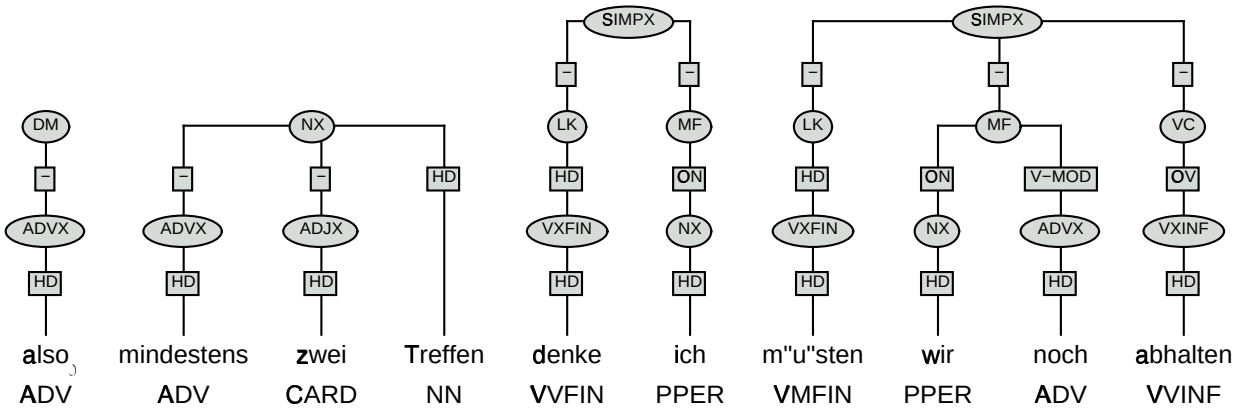


Figure 4: An example for a sentence for which *MBP* could only find a partial analysis.

| trie-based search only | |
|---|---|
| labeled recall (synt. categories) | 72.02% |
| labeled precision (synt. categories) | 77.97% |
| $F_1$ | 74.88 |
| labeled recall (incl. func. categories) | 42.65% |
| labeled precision (incl. func. categories) | 46.14% |
| $F_1$ | 44.33 |
| func. recall of attached constituents | 97.91% |
| func. precision of attached constituents | 97.90% |

| backing-off module only | |
|---|---|
| labeled recall (synt. categories) | 82.95% |
| labeled precision (synt. categories) | 87.96% |
| $F_1$ | 85.38 |
| labeled recall (incl. func. cat.) | 70.52% |
| labeled precision (incl. func. cat.) | 74.73% |
| $F_1$ | 72.56 |
| func. recall of attached const. | 94.63% |
| func. precision of attached const. | 94.51% |

Table 2: The results for the trie-based search in comparison to the backing-off module.

| | |
|---|---|
| recall (syntactic categories) | 85.15% |
| precision (syntactic categories) | 89.34% |
| $F_1$ | 87.19 |
| recall (incl. functional categories) | 76.00% |
| precision (incl. functional categories) | 79.65% |
| $F_1$ | 77.78 |
| functional recall of attached constituents | 96.56% |
| functional precision of attached constituents | 96.48% |
| $F_1$ | 96.52 |

Table 3: *MBP*'s results in the leave-one-out test with 5 000 cycles.

based on a multitude of different types of information. The types of information that are presently used are information on the word sequences themselves, their POS tags, and their chunk analysis.

The results of 79.65% correct constituents and grammatical functions in overall trees and of 96.48% correct grammatical functions for attached constituents validate the general approach of a memory-based parser. We anticipate further improvements by including morphological information (cf. e.g. (Hinrichs & Trushkina 02b; Hinrichs & Trushkina 02a)). Disambiguated morphological information for phrases is a reliable source for disambiguating the grammatical function of the phrase. Until recently, however, there was no automatic morphological disambiguation available for German.

We also plan to extend our parser to a $k - nn$ approach similar to the one followed by (Streiter 01). In standard MBL, selecting a higher number of the $k$ nearest neighbors used for classification often improves performance. For memory-based parsing, however, a higher number of $k$ would lead to several competing tree structures. For this reason, Streiter's algorithm selects a number of candidates for the most similar sentence and then uses a more fine grained search to select the ultimate candidate.

## References


(Abney 91) Steven Abney. Parsing by chunks. In Robert Berwick, Steven Abney, and Caroll Tenney, editors, *Principle-Based Parsing*. Kluwer Academic Publishers, Dordrecht, 1991.

(Abney 96) Steven Abney. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4):337 – 344, 1996.

(Black *et al.* 91) Ezra Black, Steven Abney, Dan Flickinger, Claudia Gdaniec, Ralph Grishman, Philip Harrison, Donald Hindle, Robert Ingria, Frederick Jelinek, Judith Klavans, Mark Liberman, Mitchell Marcus, Salim Roukos, Beatrice Santorini, and Tomek Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the DARPA Speech and Natural Language Workshop 1991*, Pacific Grove, CA, 1991.

(Black *et al.* 92) Ezra Black, Frederick Jelinek, John Lafferty, David Magerman, Robert Mercer, and Salim Roukos. Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Harriman, NY, 1992.

(Bod & Scha 96) Rens Bod and Remko Scha. Data-oriented language processing: An overview. Technical Report LP-96-13, Institute for Logic, Language and Computation, University of Amsterdam, 1996.

(Bod 98) Rens Bod. *Beyond Grammar: An Experience-Based Theory of Language*. CSLI Publications, Stanford, CA, 1998.

(Brants 00) Thorsten Brants. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing, ANLP-2000*, Seattle, WA, April 2000.

(Buchholz 02) Sabine Buchholz. *Memory-Based Grammatical Relation Finding*. Unpublished PhD thesis, University of Tilburg, The Netherlands, 2002.

(Charniak 01) Eugene Charniak. Immediate head parsing for language models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 116 – 123, Toulouse, France, 2001.

(Collins 99) Michael Collins. *Head-Driven Statistical Models for Natural Language Parsing*. Unpublished PhD thesis, University of Pennsylvania, 1999.

(Daelemans *et al.* 99) Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. Forgetting exceptions is harmful in language learning. *Machine Learning: Special Issue on Natural Language Learning*, 34:11 – 43, 1999.

(Herling 21) Simon Heinrich Adolf Herling. Über die Topik der deutschen Sprache. In *Abhandlungen des frankfurterischen Gelehrtenvereins für deutsche Sprache*, pages 296 – 362, 394. Frankfurt/M., 1821. Drittes Stück.

(Hinrichs & Trushkina 02a) Erhard W. Hinrichs and Julia Trushkina. Getting a grip an morphological disambiguation. In *Proceedings of KONVENS 2002*, pages 59 – 66, Saarbrücken, Germany, 2002.

(Hinrichs & Trushkina 02b) Erhard W. Hinrichs and Julia S. Trushkina. Forging agreement: Morphological disambiguation of noun phrases. In Erhard Hinrichs and Kiril Simov, editors, *Proceedings of the First Workshop on Treebanks and Linguistic Theory (TLT 2002)*, pages 78 – 95, Sozopol, Bulgaria, 2002.

(Hinrichs *et al.* 00a) Erhard W. Hinrichs, Julia Bartels, Yasuhiro Kawata, Valia Kordoni, and Heike Telljohann. The Tübingen treebanks for spoken German, English, and Japanese. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin, 2000.

(Hinrichs *et al.* 00b) Erhard W. Hinrichs, Julia Bartels, Yasuhiro Kawata, Valia Kordoni, and Heike Telljohann. The Verbmobil treebanks. In *5. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2000)*, pages 107 – 112, Ilmenau, Germany, 2000.

(Höhle 86) Tilman Höhle. Der Begriff "Mittelfeld", Anmerkungen über die Theorie der topologischen Felder. In *Akten des Siebten Internationalen Germanistenkongresses 1985*, pages 329 – 340, Göttingen, 1986.

(Kübler & Hinrichs 01a) Sandra Kübler and Erhard W. Hinrichs. From chunks to function-argument structure: A similarity-based approach. In *Proceedings of ACL/EACL 2001*, pages 338 – 345, Toulouse, France, 2001.

(Kübler & Hinrichs 01b) Sandra Kübler and Erhard W. Hinrichs. TüSBL: A similarity-based chunk parser for robust syntactic processing. In *Proceedings of the First International Human Language Technology Conference, HLT-2001*, San Diego, CA, March 2001.

(Kübler 02) Sandra Kübler. *Memory-Based Parsing of a German Corpus*. Unpublished PhD thesis, Seminar für Sprachwissenschaft, Universität Tübingen, 2002. Version of 3rd Nov. 2002.

(Stanfill & Waltz 86) Craig Stanfill and David L. Waltz. Towards memory-based reasoning. *Communications of the ACM*, 29(12):1213 – 1228, 1986.

(Stegmann *et al.* 00) Rosmary Stegmann, Heike Telljohann, and Erhard W. Hinrichs. Stylebook for the German Treebank in VERBMOBIL. Technical Report 239, Verbmobil, 2000.

(Streiter 01) Oliver Streiter. Recursive top-down fuzzy match, new perspectives on memory-based parsing. In *Proceedings of the 15th Pacific Asia Conference on Language, Information and Computation, PACLIC 2001*, Hong Kong, 2001.

(Tjong Kim Sang 01) Erik F. Tjong Kim Sang. Transforming a chunker to a parser. In Walter Daelemans, Khalil Sima'an, Jorn Veenstra, and Jakub Zavrel, editors, *Computational Linguistics in the Netherlands 2000*, pages 177 – 188. Rodopi, Amsterdam, 2001.

(Weiss & Kulikowski 91) Sholom M. Weiss and Casimir A. Kulikowski. *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. Morgan Kaufmann, San Mateo, CA, 1991.