A Hybrid Architecture for Robust Parsing of German

Erhard W. Hinrichs, Sandra Kübler, Frank H. Müller, Tylman Ule

Seminar für Sprachwissenschaft
Universität Tübingen
Wilhelmstr. 113
72074 Tübingen
Germany
{eh,kuebler,fhm,ule}@sfs.uni-tuebingen.de

Abstract

This paper provides an overview of current research on a hybrid and robust parsing architecture for the morphological, syntactic and semantic annotation of German text corpora. The novel contribution of this research lies not in the individual parsing modules, each of which relies on state-of-the-art algorithms and techniques. Rather what is new about the present approach is the combination of these modules into a single architecture. This combination provides a means to significantly optimize the performance of each component, resulting in an increased accuracy of annotation.

1. Introduction

This paper provides an overview of current research on a hybrid and robust parsing architecture for the morphological, syntactic and semantic annotation of German text corpora.

Annotation proceeds incrementally, starting with tokenization, named entity recognition, and morpho-syntactic tagging. Syntactic annotation proceeds in two steps:

1. Individual phrases are recognized by a finite-state "chunk" parser (in the sense of Abney (1996b); Aït-Mokhtar and Chanod (1997)), and

2. attachment of individual phrases into complete trees for sentential structures (including annotation of grammatical functions) is achieved by a memory-based parser (in the sense of Daelemans et al. (1999b); Daelemans et al. (1999a)).

The novel contribution of this research lies not in the individual parsing modules, each of which relies on state-of-the-art algorithms and techniques. Rather what is new about the present approach is the combination of these modules into a single architecture. This combination provides a means to significantly optimize the performance of each component, resulting in an increased accuracy of annotation. The optimization is achieved by robust heuristics for error detection of the parsing output of previous modules.

2. POS Tagging

Part-of-speech (POS) tagging nowadays is a well-known robust technique used to annotate unrestricted text with morpho-syntactic information. The task of POS tagging is defined by a set of POS tags accompanied by guidelines that determine their application. In the present framework, the POS tags are defined by the STTS German POS tagset containing 54 different tags (Schiller et al., 1995).

In the past decade, a number of different approaches for POS tagging have been implemented and evaluated, including rule-based, trigram, and maximum entropy taggers. Also, methods have been developed to combine the output of several taggers in order to improve overall results of POS tagging (Borin, 2000; van Halteren et al., 1998). In the current framework, errors of morpho-syntactic annotation are reduced along these lines by following a tagging-by-committee strategy that compares and assigns weighted probabilities to the output of several POS taggers for German, which vary in the method they apply, and also in training data. Currently, the TnT trigram tagger and the Brill rule-based tagger are used (Brants, 2000; Brill, 1992), and also two hand-crafted rule-based taggers specialized in correcting certain error types (see Section 3.). The system uses taggers that are trained separately with manually annotated news texts (315.000 tokens), with novels (150.000 tokens), and with all texts available (490.000 tokens), so that taggers can be preferred that resemble the input text more closely. In sentence 1, e.g., a tagger trained with (possibly similar, but not identical) novel texts chooses the correct POS tag `PTKVZ` (separable verb affix, chosen with 93% certainty) as opposed to the tagger trained with news texts that chooses the preposition tag `APPR` (60% certainty), although the latter has more training data available.

(1) ..., es ödete mich einfach an/PTKVZ, schon
..., it bored me simply , already
wieder in ein Flugzeug zu steigen, ...
again in a plane to climb, ...

'I was just bored boarding a plane again.'

Following the strategies outlined in van Halteren et al. (1998), the best POS tag is selected by simple majority voting extended by taking into account not only the number of taggers voting for each POS, but also the weights that some taggers assign to their choices. The POS tagging step results in a ranked sequence of POS tags, which is recorded in the linguistic markup for each word form token of an input text, so that later steps may access POS information in any required detail.

Figure 1 shows the POS tags as they are encoded in XML for the affix *an* and for the preceding word. The parts of speech are ranked by the `all` tagger trained on all texts

```
<t f='einfach'>
  <P t='ADV' r='1' c='0.6326315'>
    <j n='novel' c='0.544247'/>
    <j n='all' c='0.721016'/>
    <j n='news' c='0.7179919'/>
  </P>
  <P t='ADJD' r='2' c='0.3673685'>
    <j n='novel' c='0.455753'/>
    <j n='all' c='0.278984'/>
    <j n='news' c='0.2820081'/>
  </P>
</t>
<t f='an'>
  <P t='PTKVZ' r='1' c='0.694730185726474'>
    <j n='novel' c='0.928632'/>
    <j n='all' c='0.460828'/>
    <j n='news' c='0.4041551'/>
  </P>
  <P t='APPR' r='2' c='0.305269814273526'>
    <j n='novel' c='0.0713676'/>
    <j n='all' c='0.539172'/>
    <j n='news' c='0.5958449'/>
  </P>
</t>
```

Figure 1: XML encoding of ranked POS analyses

and the `novel` tagger trained on novels[1]. The example text is known to be text from a novel, so that for voting, the `news` tagger is ignored, resulting in a preference for the correct tag. Morphological information and syntactic structure are not shown.

3. Shallow Parsing

Chunk parsing is by now a standard technique to effectively and reliably pre-structure language data for further linguistic annotation. Non-recursive phrases (chunks) are recognized using syntactic restrictions of the composition of the chunks before attachment problems and verb-argument structure are tackled with more powerful mechanisms. Our system recognizes both simplex and complex chunks, the definition of recursion being that chunks may not contain chunks of the same type (see Figure 2). Complex chunks are defined as chunks which contain other chunks; chunks which are contained in no other chunk are called maximal chunks (Abney, 1996a).

The shallow parsing system relies on a deterministic finite-state grammar for syntactic annotation. In fact, deterministic processing is crucial for the efficiency of the entire shallow parsing approach. This determinism is guaranteed by invoking a longest-match strategy for the input to finite-state transduction at each level of annotation and by leaving unresolved many of the attachment decisions that notoriously introduce structural ambiguities. The longest-match strategy is psycholinguistically well motivated and produces the correct result in most cases (in English, e.g., this is especially true of noun-noun compounding). Our system uses the TTT suite of tools available from the LTG Edinburgh (Grover et al., 1999). The tool *fsgmatch*, which is part of the TTT suite, applies finite-state grammars to sequences of XML elements, turning the sequence of elements into a tree structure. In the current framework, this XML tree is used directly to encode the linguistic tree structure resulting from shallow parsing.

Systems using chunk parsing typically work with a bottom-up strategy, which recognizes chunks before sub-

[1]The example is not part of the training data.

```
[PC
    .APPR    aus                              from
    [NC
        .PPOSAT  ihrem                        their
        [AJACC
            [AJAC
                [AVC
                    .ADV     so ]             ever so
                .ADJA    gewöhnlichen ]       trivial
            .$,      ,
            [AJAC
                [AVC
                    .ADV     so ]             ever so
                .ADJA    grauen ]             grey
            .$,      ,
            [AJAC
                [AVC
                    .ADV     so ]             ever so
                .ADJA    tristen ] ]          dull
        .NN      Dasein ] ]                   existence
[PC
    .APPR    in                               in
    [NC
        .ART     der                          the
        .NN      DDR-Provinz ] ]              GDR-backwaters
```

'from their ever so trivial, ever so grey, ever so dull existence in the backwaters of the GDR'

Figure 2: Two complex maximal chunks

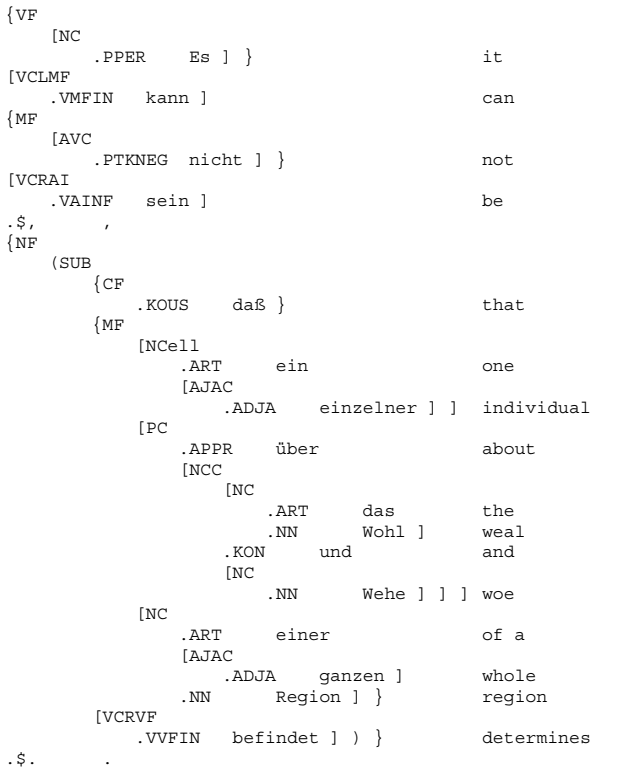
clauses and sentences are matched. Our system, by contrast, takes advantage of top-down information provided by a characterization of German clause types in terms of *topological fields*[2]. Topological fields describe sections in the German sentence with regard to the distributional properties of the verb complex in main clauses, on the one hand, and the verb complex and the subordinator in subclauses, on the other hand. These two constituents make up the *sentence bracket* ('Satzklammer'), which is divided into a left part (LK) and a right part (RK). In main clauses, the LK contains the finite verb, and all other verbal elements are contained in the RK. In subordinate clauses, the LK contains the subordinator, and all verbal elements are contained in the RK. As can be seen in Figure 3, the LK is realized as a CF (**c**omplementizer **f**ield) in subordinate clauses or as a VCL_ (**v**erb **c**omplex **l**eft part)[3] in main clauses. The RK is realized as a VCR_ (**v**erb **c**omplex **r**ight part) in all types of clauses. The RK is optional in main clauses (see Figure 4).

After the annotation of the sentence bracket, the following topological fields can be described relative to it: The section before the LK is called the initial field (VF), the section in between the LK and the RK is called the middle field (MF) and the section following the RK is called the final field (NF). Figure 3 gives an example in which all three fields are realized. The section before *kann* is annotated as VF, the section in between *kann* and *sein* is annotated MF and the section after *sein* is annotated NF. If two clauses are coordinated, the coordinator is contained in a coordinator field (KOORDF) (see Figure 4).

The composition of the topological field structure in a

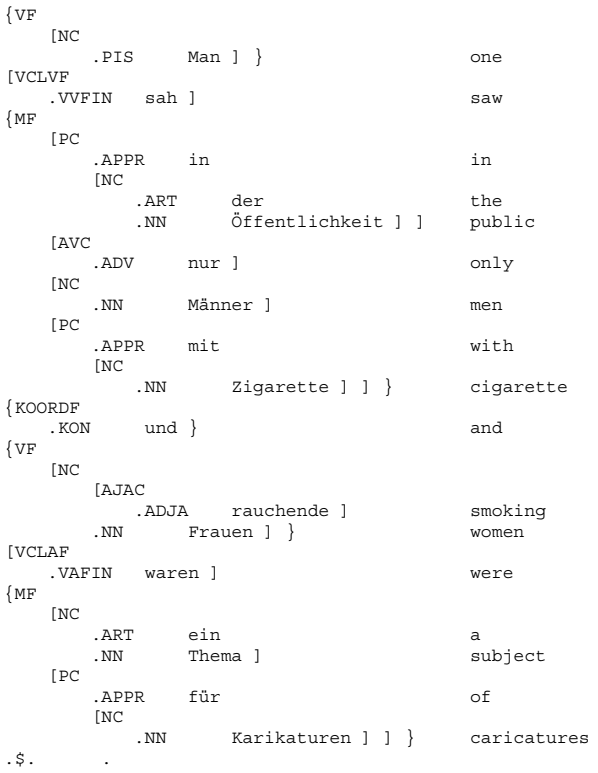
[2]The characterization of German clause types and corresponding regularities of word order in terms of topological fields has a long tradition in empirical investigations of German syntax (Herling, 1821; Erdmann, 1886; Drach, 1937; Reis, 1980; Höhle, 1985) and is by now widely accepted as a theory-neutral classification of German clauses and their internal structure.

[3]The letters after the VCL_ (and VCR_ respectively) denote the types of verbs contained in the verb complex.



'It is totally unacceptable that one idividual determines the weal and woe of a whole region.'

Figure 3: Subclause embedded in NF of main clause



'In public, you could see only men with cigarettes and smoking women were a subject of caricatures.'

Figure 4: Ambiguous scope of coordination

clause is subject to syntactic restrictions. These syntactic restrictions can be compared to the syntactic restrictions in chunks in that they do not depend on the lexical entry of the tokens but are universally valid for all tokens of one POS tag class. The structure of topological fields discloses the borders and the composition of a clause and thus reveals the whole anatomy of the sentence. Topological fields and clauses together with chunks provide a solid shallow preanalysis of a sentence.

By annotating topological fields and basic clause structure first, attachment and coordination ambiguities are effectively reduced even before chunking takes place. Thus, our parser employs a mixed top-down, bottom-up control regime for the incremental linguistic annotation of topological fields and clauses, first, and chunks, afterwards. A similar strategy has already been used to pre-structure sentences for an information retrieval system (Neumann et al., 2000; Neumann and Piskorski, 2002). Figure 4 shows an example of such a pre-structured analysis. If chunking had been done before field analysis, it would not have been clear whether the string *Männer mit Zigaretten und rauchende Frauen* was a coordinated noun phrase. With the pre-structuring, this reading can be ruled out, thus reducing coordination ambiguity. Figure 4 also shows that, after the pre-structuring, the search space for the annotation of chunks has become smaller thus speeding up the parser. While, before the pre-structuring took place, the search space was the whole sentence, the search space after the pre-structuring is the topological field. Another advantage is that, with the pre-structuring, the scope for the arguments of the verb is considerably reduced because the potential sites of the arguments of the verb are limited by the topological fields which can be assigned to a verb. Thus, e.g. in Figure 5 the arguments of *isolieren* can only be contained in the MF of the subclause and the arguments of *gewinnen* can only be in the MF of the main clause (as regards phrasal arguments) or in the VF of the main clause (as regards a clausal argument).

In addition, the shallow parser, which is used for the first level of syntactic annotation, is utilized for the correction of tagging errors which are known to have a particularly negative effect on parsing accuracy for German. Two classes of common tagging errors in German concern the distinction between finite and non-finite verb forms and the distinction between homonymous prepositions and subordinators. These errors can be corrected by employing a mixed control regime of top-down and bottom-up shallow parsing. Utilizing top-down information about the macrostructure of German clause types as it is reflected in their topological structure, it becomes possible to detect missing subordinators and finite verbs, which at the POS tagging level were wrongly tagged as prepositions and non-finite verbs, respectively.

This mechanism is used in such cases in which the parser is not able to assign any grammatical structure to a given POS tag sequence. If there is no parsable POS tag sequence, the parser makes use of the ranked POS tag assignments. The parser considers the second-best tag and tries to match a parsable sequence again. Provided that the parser succeeds, the second-best tag is changed into the best

```
{VF
    (SUB
        {CF
            .APPR --> KOUS   Seit }        since
        {MF
            [NCC
                [NC
                    .NE      Banting ]     Banting
                .KON     und               and
                [NC
                    .NE      Best ] ]      Best
            [NC
                .NN      Insulin ]         insulin
            [PC
                .APPRART zum               for the
                [AVC
                    .ADV     erstenmal ] ] } first time
        [VCRMFVI
            .VVINF   isolieren             isolate
            .VMFIN   konnten ] )           could
    .$,      , }
[VCLAF
    .VAFIN   haben ]                       have
{MF
    [NC
        .ART     die                       the
        .NN      Mediziner ]               physicians
    [NC
        [AJAC
            .ADJA    lebenserhaltende ]    life-preserving
        .NN      Kontrolle ]               control
    [PC
        .APPR    über                      of
        [NC
            .NN      Diabetiker ] ] }      diabetics
[VCRMIVI
    .VVINF   gewinnen                      win
    .VMINF   können ]                      could
.$.      .
```

'Ever since Banting and Best have been able to isolate insulin for the first time, physicians have been able to win life-preserving control of diabetics.'

Figure 5: Ambiguous subordinator (*seit*)

tag and the whole POS sequence is annotated. This strategy thus uses linguistic knowledge already encoded in the parser of our annotation system and in the annotation itself to refine POS tagging. The strategy is resemblant of the one described in Hirakawa et al. (2000). Figure 5 gives an example: The token *seit* is ambiguous in that it can be either a preposition (APPR) or a subordinator (KOUS). However, as the system first tries to match topological fields, the parser would fail to assign a correct structure if the token was tagged as a preposition because the RK requires a CF with a subordinator to appear in sentence-initial position. The parser then tries to match the structure with the second-best tag (KOUS) and annotates the structure. The same mechanism works with the finite vs. non-finite (VVFIN vs. VVINF) ambiguity of many verbs (See Figure 6, where *nehmen* is ambiguous and was first tagged VVFIN but annotating the structure only works with *nehmen* as a non-finite verb (VVINF) because *kann* is the finite verb in the clause and it requires a non-finite verb.).

4. Memory-Based Parsing

As mentioned in the previous section, chunk parsing in conjunction with the descriptive and predictive power of the topological fields model for characterizing German sentence structure provides an effective way of isolating and annotating major syntactic constituents and of correcting tagging errors introduced by the POS tagger. However, the chunk parser is not immune from producing wrong results,

```
{VF
    [NC
        .NE      Libyen ] }                Libya
[VCLMF
    .VMFIN   kann ]                        can
{MF
    [NC
        .PIAT    keinen                    no
        .NN      Einfluss ]                influence
    [PC
        .APPR    auf                       on
        [NC
            .ART     die                   the
            .NN      Politik ] ]           politics
    [NC
        .NE      Marokkos ] }              of Morocco
[VCRVI
    .VVFIN --> VVINF   nehmen ]            exert
.$.      .
```

'Libya can exert no influence on the politics of Morocco.'

Figure 6: Ambiguous non-finite verb (*nehmen*)

especially for non-local dependencies. A common source of errors of this sort are coordination structures for which, in accordance with the longest-match strategy, coordination of adjacent NPs is wrongly favored in cases where sentence coordination would have been the correct structure and where structuring the sentence into topological fields does not provide conclusive information about the scope of the coordination.

In addition, a chunk parser provides only a partial syntactic analysis since its main goal is the robust annotation of unrestricted text or transliterated speech. As a consequence, dependency relations between individual chunks, such as grammatical functions or modification relations, within a clause remain unspecified. However, for many NLP applications, the correct determination of such relations is indispensable.

In order to provide such deeper and more complete syntactic annotation, the chunk parser output is processed further by a second parsing component, which employs a memory-based parsing strategy.

Memory-based learning (Stanfill and Waltz, 1986; Aha et al., 1991) has been applied previously to a variety of classification tasks in natural processing, including grapheme-phoneme conversion (Stanfill and Waltz, 1986; van den Bosch and Daelemans, 1993), part-of-speech tagging (Cardie, 1993; Daelemans et al., 1996), word sense disambiguation (Escudero et al., 2000; Veenstra et al., 2000) or PP attachment (Buchholz, 1998). Applying such techniques for the purposes of inducing syntactic trees constitutes a major challenge for such memory-based approaches since the set of grammatically well-formed trees in a given natural language is, in principle, infinite. Therefore, memory-based parsing goes beyond ordinary classification tasks for which the class of candidates is finite and of "manageable" cardinality. Part-of-speech tagging is a typical example in this regard, with basic tagsets for many languages ranging from twenty to at most two hundred distinct labels. What distinguishes syntactic annotation from such ordinary classification tasks is the fact that a finite set of morpho-syntactic labels and phrasal syntactic categories can be combined recursively to produce a potentially infinite number of syntactic structures.

The key observation that makes the application of memory-based techniques to syntactic parsing of natural languages at all feasible, is the fact that the potentially infinite set of candidate structures is in practise restricted by the finite length of the input string to be parsed. For any given input string the set of candidate structures will be finite. The parsing problem, thus, consists of choosing from an infinite set of well-formed syntactic structures the optimal (finite) structure for a given input string. Classical, rule-based parsers solve this task by factoring the problem into local decisions about local candidate substructures. (Probabilistic) context-free parsers are prime examples of such a strategy. By contrast, data-driven (Bod, 1998) or memory-based approaches to parsing make no such locality assumption. Instead, they consider substructures of arbitrary size and select those substructures for incorporation into larger trees which best fit the input data. In the case of memory-based parsing, the parsing algorithm retrieves the most similar parsing tree from stored training examples (i.e. from a treebank) by using the results of the previous annotation steps as features for the similarity metric. This tree is then adapted in a second step to match the input sentence. Utilizing the complete sentence as context and retrieving the complete tree in one step ensures that the decision is based on the highest amount of information possible and that the full parse is also achieved deterministically. A more detailed description of the algorithm can be found in Kübler and Hinrichs (2001a) and Kübler and Hinrichs (2001b).

The division of labor between the chunking and tree construction modules can best be illustrated by an example. For complex sentences such as the German input *wie würde Ihnen denn der Termin passen, am Mittwoch den zehnten und am Donnerstag den elften November*, the chunk parser produces a structure in which some constituents remain unattached or partially annotated in keeping with the chunk parsing strategy to factor out recursion and to resolve only unambigous attachments, as shown in Figure 7.

In the case at hand, the subconstituents of the extraposed coordinated prepositional phrase are not attached to the simplex clause that ends with the non-finite verb that is typically in clause-final position in declarative main clauses of German. Moreover, each conjunct of the coordinated prepositional phrase consists of a base prepositional chunk and separate noun chunk which needs to be attached as an apposition to the noun phrase within the prepositional phrase. The memory-based parsing module enriches the chunk output as shown in Figure 8[4]. Here, the complex PP phrases have been coordinated and integrated correctly into the clause as a whole. In addition, function labels such as *v-mod* (for: verbal modifier), *hd* (for: head), *od* (for: dative object), *mod* (for: ambiguous modifier), *on* (for: subject), *ov* (for: verbal object), and *app* (for: apposition) have been added that encode the function-argument structure of the sentence.

Apart from constructing complete tree structures on

[4]The trees in Figure 8 and in Figure 10 follow the data format for trees defined by the NEGRA project of the Sonderforschungsbereich 378 at the University of the Saarland, Saarbrücken. They were printed by the NEGRA annotation tool (Brants and Skut, 1998).

```
{VF
    .PWAV    wie }                          how
[VCLAF
    .VAFIN   würde ]                        would
{MF
    [NC
        .PPER    Ihnen ]                    you
    [AVC
        .ADV     denn ]                     then
    [NC
        .ART     der                        the
        .NN      Termin ] }                 appointment
[VCRVI
    .VVINF   passen ]                       suit
{NF
    [PC
        .APPRART am                         on the
        [NC
            .NN      Mittwoch ] ]           Wednesday
    [NCell
        .ART     den                        the
        [AJAC
            .ADJA    zehnten ] ]            tenth
    .KON     und                            and
    [PC
        .APPRART am                         on the
        [NC
            .NN      Donnerstag ] ]         Thursday
    [NC
        .ART     den                        the
        [AJAC
            .ADJA    elften ]               eleventh
        .NN      November ] }               November
```

'How would the appointment suit you on Wednesday tenth and on Thursday eleventh of August.'

Figure 7: A complex sentence parsed by the chunk parser

```
{VF
    [NC
        .PDS     das ] }                    this
[VCLAF
    .VAFIN   ist ]                          is
{MF
    [NCC
        [NC
            .ART     ein                    a
            .NN      Freitag ]              Friday
        .KON     und                        and
        [NC
            .PPER    wir ] ] }              we
[VCRVP
    .VVPP    wissen ]                       know
{NF
    (SUB
        {CF
            .KOUS    daß }                  that
        {MF
            [NC
                .PPER    Sie ]              you
            [NC
                .NE      Piano ]            piano
            [AVC
                .ADV     sehr ] }           a lot
        [VCRMF
            .VMFIN   mögen ] ) }            like
```

'This is a Friday and we know that you like the piano a lot.'

Figure 9: Wrongly coordinated NP chunks

the basis of pre-chunked input, the memory-based parsing component is also used for correcting errors introduced by the chunk parser. As mentioned before, the chunk parser in accordance with the longest-match strategy sometimes wrongly favors coordination of adjacent NPs in cases where sentence coordination would have been the correct structure. The sentence in Figure 9 is a typical example of this kind. Instead of chunking the pronoun *wir* as part of the second conjunct of a sentence coordination structure, it is

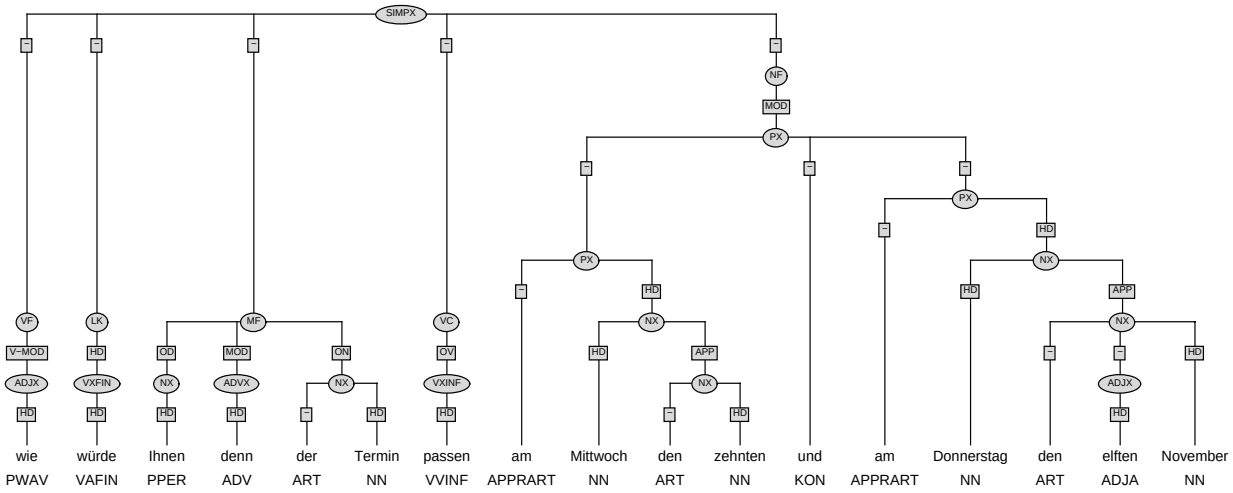


Figure 8: Output of the memory-based parser

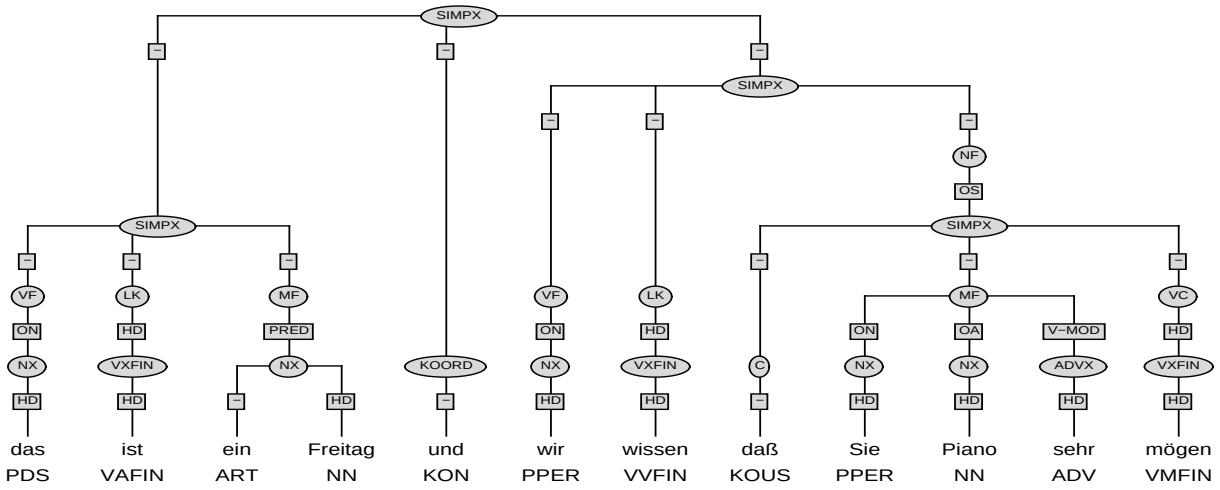


Figure 10: Corrections of the memory-based parser

incorrectly grouped with the preceding NP *Freitag* as a coordinated NP. Such an error occurs since the chunk parser typically assigns structure on the basis of the local context of a word or phrase. For the sentence in Figure 9, this local context to the right of *wir* consists of a verb that was erroneously tagged as a past participle, which is a clear indication of a right bracket. Since the resulting POS pattern is valid for German, the tagging error could not be detected and corrected by the chunk parser. The memory-based parser, however, takes into account the global syntactic structure assigned to previously seen instances. Thus, it has a better chance of producing the correct constituent structure for such non-local phenomena. Accordingly, in the tree structure shown in Figure 10, that is produced by the memory-based parser, the chunking error has been corrected and the correct sentential coordination has been assigned.

5. Conclusion

The above parsing scheme has been used for the syntactic annotation of the VERBMOBIL corpus of spoken German (Hinrichs et al., 2000; Stegmann et al., 2000) and the German reference corpus (DEREKO, 2002) of written texts. The resulting robust annotations can be used by theoretical linguists, who are interested in large-scale, empirical data, and by computational linguists, who are in need of training material for a wide range of language technology applications. The usability of the annotated corpora is further enhanced by an XML encoding at each level of annotation, facilitating easy searching of the data, enabling easy data conversion according to user-driven data formats, and supporting graphical visualization of the data by standard XML tools.

6. Acknowledgments

The research reported here was supported by the German Research Council (DFG) as part of the Sonderforschungsbereich 441 “Linguistische Datenstrukturen” (Linguistic Data Structures). The third author was also supported by a grant of the graduate school “Integriertes Linguistikstudium” funded by the DFG. The authors are grateful to Thorsten Brants and Eric Brill, who made their POS

taggers available to them, and to the LTG Edinburgh, who provided them with the TTT suite of tools.

7. References

Steven Abney. 1996a. Chunk stylebook. Technical report, University of Tübingen, Tübingen. Draft1: http://www.sfs.nphil.uni-tuebingen.de/∼abney/Papers.html#96i.

Steven Abney. 1996b. Partial parsing via finite-state cascades. In John Carroll, editor, *Workshop on Robust Parsing (ESSLLI '96)*, pages 8 – 15, Prague, Czech Republic.

David Aha, Dennis Kibler, and Marc K. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37 – 66.

Salah Aït-Mokhtar and Jean-Pierre Chanod. 1997. Incremental finite-state parsing. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP 1997)*, pages 72 – 79, Washington, D.C.

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

Lars Borin. 2000. Something borrowed, something blue: Rule-based combination of POS taggers. In *Second International Conference on Language Resources and Evaluation (LREC 2000)*, pages 21–26, Athens, 31 May – 2 June.

Thorsten Brants and Wojciech Skut. 1998. Automation of treebank annotation. In *Proceedings of NeMLaP-3/CoNLL98*, pages 49 – 57, Sydney, Australia.

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

Eric Brill. 1992. A simple rule-based part-of-speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP 1992)*, pages 152–155, Trento, Italy.

Sabine Buchholz. 1998. Distinguishing complements from adjuncts using memory-based learning. In *Proceedings of the ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing*, pages 41 – 48.

Claire Cardie. 1993. Using decision trees to improve case-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 25 – 32. Morgan Kaufmann.

Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. 1996. MBT: A memory-based part of speech tagger-generator. In Eva Ejerhed and Ido Dagan, editors, *Proceedings of the Fourth Workshop on Very Large Corpora*, pages 14 – 27, Copenhagen.

Walter Daelemans, Sabine Buchholz, and Jorn Veenstra. 1999a. Memory-based shallow parsing. In *Proceedings of CoNLL-99*, pages 53 – 60, Bergen, Norway.

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

DEREKO. 2002. DEREKO: The German Reference Corpus Project. http://www.sfs.nphil.uni-tuebingen.de/dereko/.

Erich Drach. 1937. *Grundgedanken der Deutschen Satzlehre*. Frankfurt/M.

Oskar Erdmann. 1886. *Grundzüge der deutschen Syntax nach ihrer geschichtlichen Entwicklung dargestellt*. Stuttgart. Erste Abteilung.

Gerard Escudero, Lluìs Márquez, and German Rigau. 2000. Naive Bayes and exemplar-based approaches to word sense disambiguation revisited. In *Proceedings of the 14th European Conference on Artificial Intelligence, ECAI'2000*, pages 421 – 425, Berlin.

Claire Grover, Colin Matheson, and Andrei Mikheev, 1999. *TTT: Text Tokenisation Tool*. Language Technology Group, University of Edinburgh, Edinburgh. http://www.ltg.ed.ac.uk/software/ttt/tttdoc.html.

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

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

Hideki Hirakawa, Kenji Ono, and Yumiko Yoshimura. 2000. Automatic Refinement of a POS Tagger Using a Reliable Parser and Plain Text Corpora. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken. International Committee on Computational Linguistics ICCL.

Tilman N. Höhle. 1985. Der Begriff 'Mittelfeld'. Anmerkungen über die Theorie der topologischen Felder. In A. Schöne, editor, *Kontroversen alte und neue. Akten des 7. Internationalen Germanistenkongresses Göttingen*, pages 329–340.

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

Sandra Kübler and Erhard W. Hinrichs. 2001b. 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.

Günter Neumann and Jakub Piskorski. 2002. A shallow text processing core engine. Under review by the Journal of Computational Intelligence; submitted in April 2000; http://www.dfki.de/∼neumann/publications/new-ps/comp-intell.pdf.

Günter Neumann, Christian Braun, and Jakub Piskorski. 2000. A divide-and-conquer strategy for shallow parsing of German free texts. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP 2000)*, pages 239–246, Seattle, Washington.

Marga Reis. 1980. On justifying topological frames: 'Positional field' and the order of nonverbal constituents in German. *DRLAV: Revue de Linguistique*, 22/23:59–85.

Anne Schiller, Simone Teufel, and Christine Thielen. 1995. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Universität Stuttgart, Institut

für maschinelle Sprachverarbeitung, and Seminar für Sprachwissenschaft, Universität Tübingen.

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

Rosmary Stegmann, Heike Telljohann, and Erhard W. Hinrichs. 2000. Stylebook for the German Treebank in VERBMOBIL. Technical Report 239, Verbmobil.

Antal van den Bosch and Walter Daelemans. 1993. Data-oriented methods for grapheme-to-phoneme conversion. In *Proceedings of the Sixth Conference of the European Chapter of the ACL*, pages 45 – 53.

Hans van Halteren, Jakub Zavrel, and Walter Daelemans. 1998. Improving data driven wordclass tagging by system combination. In *COLING-ACL*, pages 491–497.

Jorn Veenstra, Antal van den Bosch, Sabine Buchholz, Walter Daelemans, and Jakub Zavrel. 2000. Memory-based word sense disambiguation. *Computers and the Humanities, Special Issue on Senseval, Word Sense Disambiguations*, 34(1/2):171 – 177.